An Expert's Guide To Oracle: The Best Essays of 2005

Lewis R Cunningham

January 21, 2006

Essays previously appeared on the Expert's Guide to Oracle Technology web log on ItToolbox.com. All essays have been edited for content and formating in this document.

Jackson Square, New Orleans, LA - Christmas Day 2005

Table of Contents

Chapter 1 - Introduction

***It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity .* - Charles Dickens, A Tale of Two Cities.**

2005, oh what a year! I will forever call it the year of the Oracle blog. It's the year I started blogging but more importantly, it's the year dozens of developers, DBAs and other Oracle dweebishes started blogging. Of course being one, I mean dweebish in the best possible way.

But the year wasn't all about blogs. It was the year I decided I would start presenting. We're in 2006 as I write this and I haven't presented yet but next Tuesday I will have my first conference under my belt (Desktop Conference 2006). Am I crazy? I haven't yet done my first and I already have a presentation scheduled for May and another in June.

It was also a year of tragedies: a tsunami in the pacific, Katrina for the gulf coast, floods in New England, fires in Texas, California and Oklahoma, earthquakes in several countries. Fear was all the rage: terrorists, bird flu, war.

Katrina impacted my family pretty deeply. As you read through this document, you'll see some pictures of my post-katrina visit home to New Orleans at Christmas 2005. My wife took these pictures (she told me to say "beloved wife", yes dear). My mom and sisters are living in trailers. They're the lucky ones. A lot of families are still living in tents. If you would like to contribute to help people on the gulf coast, you can go to the Charity Navigator.

I wrote some of these essays while visiting New Orleans for the Oracle Developer Tools User Group (ODTUG). I sat at my mother's kitchen table and then logged on using my work account to post them to my blog. That kitchen table was destroyed in the flooding following Katrina as was the desk that held the computer I gave her. She managed to save the computer (by putting it up high when the water started rising) but as of six months post-katrina, she still doesn't have a place to hook it back up and use it.

In those same essays, I complain that I had to pay $2.09 for a gallon of gas. Yesterday I paid $2.25 and was glad that it has gotten cheaper recently. What a difference a single year makes.

I guess that's enough of me; let's move on to what you will find below. I chose my favorite blog entries for this document. Some generated a lot of comments, some generated email, some got responses from other bloggers, some I just like.

In a few cases, I have included user comments. Some of the included comments I found interesting, some pointed out pertinent facts and some disagreed with me. I didn't post the majority of comments. I can only blame that on space and time. I was torn between listing comments that corrected typos and such and finally decided to correct the text of the entry instead. If you have corrected me in the past, thank you. I do appreciate it.

And for those of you who chose to download this and read to this point, thank you. I tried to format this so that it is a little bit easier to read, print and pass to friends and co-workers than the blog is. HTML kind of sucks for printing and passing around. Sending URLs sure is handy though.

If you enjoy what's in here, make sure you subscribe to my blog at An Expert's Guide to Oracle Technology and as always, if you have question or concerns, write to me at lewisc@rocketmail.com. Put "blog" in the subject or the spam monkey might eat it.

Take care,

LewisC
02/16/2006

Update: I wrote this and never published it. It's now a year since Katrina and my mom is still living in a trailer. I became an Oracle ACE. I paid $2.91 for a gallon of gas. Life goes on.

LewisC
08/05/2006

French Quarter Computers, New Orleans, LA - Christmas Day 2005

Chapter 2- SQL

It's better to be last through a green light than to be first at a red light.- LewisC

Merge (AKA Upsert) - A Definition in Plain English

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/005246.asp

Posted 8/3/2005

The basic Data Manipulation Language (DML) commands that have been in use over the years are Update, Insert and Delete. They do exactly what you expect: Insert adds new records, Update modifies existing records and Delete removes records.

In version 9i, Oracle added a new DML command, MERGE. This command was improved in 10g. MERGE is generally associated with Data Warehouse and ETL but truthfully, it can be used anywhere you need to move data from one table to another. The MERGE command, also called an UPSERT, is a combination of INSERT and UPDATE.

The basic syntax of a MERGE is:

```
MERGE INTO destination_table dest
  USING (SELECT col1, col2, col3 FROM source_table) source
       ON (dest.col1 = source.col1)
       WHEN MATCHED THEN
           UPDATE SET dest.col2 = source.col2,
                                dest.col3 = source.col3
       WHEN NOT MATCHED THEN
            INSERT (dest.col1, dest.col2, dest.col3)
            VALUES (source.col1, source.col2, source.col3)
/
```

That is a basic MERGE. The WHEN MATCHED clause as well as the WHEN NOT MATCHED clause is optional. You can leave off the WHEN MATCHED if you need to insert records and there might be dupes.

MERGE INTO designates the table where the data is going. USING selects data from a table (or tables, or even a view). ON is the matching criteria. WHEN MATCHED means that a match was found (UPDATE) and WHEN NOT MATCHED a match was not found (INSERT).

In a project of mine, I MERGE records from an external table into a transaction detail table.

Here is the directory object:

```
CREATE DIRECTORY incoming_files AS '/tmp'
/
```

And the external table:

```
CREATE TABLE INCOMING_TRANSACTIONS
(
  TRAN_ID          CHAR(13 BYTE),
  TRAN_DATE        CHAR(9 BYTE),
  DL_NO            CHAR(30 BYTE),
  ZIP_CODE         CHAR(9 BYTE),
  TR_STATE         CHAR(2 BYTE),
  TRAN_AMT         CHAR(8 BYTE)
)
ORGANIZATION EXTERNAL
  (  TYPE ORACLE_LOADER
     DEFAULT DIRECTORY INCOMING_FILES
     ACCESS PARAMETERS
       ( records delimited by newline
       badfile 'incoming_transactions.bad'
       discardfile 'incoming_transactions.disc'
       logfile 'incoming_transactions.log'
        fields (
           tran_id CHAR(13),
           tran_date CHAR(9),
           dl_no CHAR(30),
           zip_code CHAR(9),
           tr_state CHAR(2),
           tran_amt CHAR(8)
      )
     )
     LOCATION (INCOMING_FILES:'incoming_transactions.dat')
  )
REJECT LIMIT UNLIMITED
/
```

This external table will read a file, incoming_transactions.dat, and present it as a table in Oracle. If you would like more information on external tables, let me know and I'll create an entry for those.

My transaction detail table looks like this:

```
CREATE TABLE transaction_details
(
  TRAN_ID               NUMBER DEFAULT -1   NOT NULL,
  TRAN_DT               DATE                NOT NULL,
  ZIP_CD                VARCHAR2(9 BYTE)    NOT NULL,
  STATE_CD              VARCHAR2(2 BYTE)    NOT NULL,
  DRIVERS_LICENSE#      VARCHAR2(40 BYTE)   NOT NULL,
  TRANSACTION_AMT       NUMBER(17,2)        NOT NULL
)
/
```

The real transactions_detail table is a range partitioned table which is partitioned by tran_dt by month. Both the external table and transaction table also have additional parameters for parallel and other options.

The incoming_transactions.dat file is replaced daily and new data is loaded. An INSERT APPEND would be the fastest way to load the data, but it possible that an existing TRAN_ID could be in the new file. It may be a mistake or a correction.

If you're following along and reproducing the code in SQL*Plus, this is a sample data record:

```
123456789012301-JAN-05X123456789Y123456789Z123456789337162204FL12345.78
```

One way to load the data prior to MERGE would be to UPDATE any records that matched on TRAN_ID and then INSERT any records that did not exist in TRANSACTION_DETAILS.

With MERGE, we would do it like this, although I'm defaulting the date format mask and any other formatting for ease of reading. Normally you would not want a date to default.

```
MERGE INTO transaction_details td
  USING (SELECT tran_id, tran_date, zip_code, tr_state, dl_no, tran_amt
           FROM incoming_transactions) new_trans
  ON (td.tran_id = new_trans.tran_id)
  WHEN MATCHED THEN
    UPDATE SET td.zip_cd = new_trans.zip_code,
               td.state_cd = new_trans.tr_state,
               td.drivers_license# = new_trans.dl_no,
               td.transaction_amt = new_trans.tran_amt,
               td.tran_dt = new_trans.tran_date
  WHEN NOT MATCHED THEN
    INSERT (td.tran_id, td.tran_dt, td.zip_cd, td.state_cd,
            td.drivers_license#, td.transaction_amt)
    VALUES (new_trans.tran_id, new_trans.tran_date, new_trans.zip_code,
            new_trans.tr_state, new_trans.dl_no, new_trans.tran_amt)
/
```

Both the update and the insert clauses allow an optional WHERE clause. It might look like this:

```
    UPDATE SET td.zip_cd = new_trans.zip_code,
               td.state_cd = new_trans.tr_state,
               td.drivers_license# = new_trans.dl_no,
               td.transaction_amt = new_trans.tran_amt,
               td.tran_dt = new_trans.tran_date
    WHERE (new_trans.zip_code != '337162204')
```

That would prevent any records from that zip code from being updated but they will still be available to the INSERT clause. If you included that WHERE clause in the USING clause select statement, that record would not be available for either the INSERT or the UPDATE.

The UPDATE clause also allows an optional DELETE statement.

```
 UPDATE SET td.zip_cd = new_trans.zip_code,
               td.state_cd = new_trans.tr_state,
               td.drivers_license# = new_trans.dl_no,
               td.transaction_amt = new_trans.tran_amt,
               td.tran_dt = new_trans.tran_date
    DELETE WHERE (new_trans.zip_code = '337162204')
```

This will delete the record rather than update it. Actually, it updates it and then deletes it. But the effect is that the record is removed. I use MERGE a lot and have not yet had a use for DELETE. Nice to know it's there if I need it though. If you have used it, let me know how and why.

And that's about it. This turned out to be shorter than I thought it would. MERGE is another great tool for the toolbox.

I am doing a Reverse Ask Tom

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006151.asp

Posted 10/13/2005

Instead of providing an interesting solution to a SQL question, I'm looking for a solution to an interesting SQL query. heh

I need the difference of the average between the count of transactions in two time periods. Let's try that again. As an example, I need the difference between the average number of transactions in the last 7 days and the last 6 months (including the last 7 days).

I have the following table:

```
CREATE TABLE transactions (
   group_id number,
   tran_dt date )
/

INSERT INTO transactions values (1, sysdate - 181);
INSERT INTO transactions values (1, sysdate - 180);
INSERT INTO transactions values (1, sysdate - 179);
INSERT INTO transactions values (1, sysdate - 178);
INSERT INTO transactions values (1, sysdate - 177);
INSERT INTO transactions values (1, sysdate - 176);
INSERT INTO transactions values (1, sysdate - 175);
INSERT INTO transactions values (1, sysdate - 174);
INSERT INTO transactions values (1, sysdate - 173);
INSERT INTO transactions values (1, sysdate - 7);
INSERT INTO transactions values (1, sysdate - 6);
INSERT INTO transactions values (1, sysdate - 5);
INSERT INTO transactions values (1, sysdate - 4);
INSERT INTO transactions values (1, sysdate - 3);
INSERT INTO transactions values (1, sysdate - 2);
INSERT INTO transactions values (1, sysdate - 1);
INSERT INTO transactions values (1, sysdate );

commit;
```

Ok, here is the basic query I've come up with:

```
select  group_id,
   tran_dt,
   count(*)  -- First it's the 7 day count
     over( order by tran_dt
           range between 7 preceding
                     and 0 following ) count7,
     count(*) -- next the 180 count
       over( order by tran_dt
             range between 180 preceding
                       and 0 following) count180,
    (count(*)
       over( order by tran_dt
              range between 7 preceding
                        and 0 following ) / 7) -
       (count(*) -- The 7 day average (cnt/days) - 180 avg
          over( order by tran_dt
               range between 180 preceding
                 and 0 following) / 180) cnt
  from transactions dt
  WHERE tran_dt -- getting just last 6 months of data
        BETWEEN trunc(add_months(trunc(sysdate), -6))
            AND trunc(sysdate)
/

  GROUP_ID TRAN_DT       COUNT7   COUNT180        CNT
---------- --------- ---------- ---------- ----------
         1 15-APR-05          1          1 .137301587
         1 16-APR-05          2          2 .274603175
         1 17-APR-05          3          3 .411904762
         1 18-APR-05          4          4 .549206349
         1 19-APR-05          5          5 .686507937
         1 20-APR-05          6          6 .823809524
         1 21-APR-05          7          7 .961111111
         1 22-APR-05          7          8 .955555556
         1 23-APR-05          7          9        .95
         1 06-OCT-05          1         10 .087301587
         1 07-OCT-05          2         11 .224603175

  GROUP_ID TRAN_DT       COUNT7   COUNT180        CNT
---------- --------- ---------- ---------- ----------
         1 08-OCT-05          3         12 .361904762
         1 09-OCT-05          4         13 .499206349
         1 10-OCT-05          5         14 .636507937
         1 11-OCT-05          6         15 .773809524
         1 12-OCT-05          7         15 .916666667

16 rows selected.
```

I think the analytics are doing what I want. If I run that, the very last record is the one I really want. I want the average of 7 transactions in the last 7 days minus the average of 15 days in the last 180 days ~= 0.917.

And this is where I get stumped. I know I can do some kludgey code with an outer query and rownum, but this really involves multiple tables and this will be one of many column expressions in the query. There will actually be several of these expressions (different time frame for the first time frame) sharing the same from and where clause. Other than the rownum trick, I am drawing a complete blank.

This is the kludge:

```
select group_id, cnt
from (
select  group_id,
   tran_dt,
   count(*)  -- First it's the 7 day count
     over( order by tran_dt
           range between 7 preceding
                     and 0 following ) count7,
     count(*) -- next the 180 count
       over( order by tran_dt
             range between 180 preceding
                       and 0 following) count180,
    (count(*)
       over( order by tran_dt
             range between 7 preceding
                       and 0 following ) / 7) -
      (count(*) -- The 7 day average (cnt/days) - 180 avg
         over( order by tran_dt
               range between 180 preceding
                 and 0 following) / 180) cnt
  from transactions dt
  WHERE tran_dt -- getting just last 6 months of data
        BETWEEN trunc(add_months(trunc(sysdate), -6))
            AND trunc(sysdate)
  order by tran_dt desc
)
where rownum = 1
/
```

It's Thursday night at 10pm and I'm going on a 4-day vacation starting about 5 hours ago. I just wanted to get this one, simple little query finished up and then go to bed. Arghhhhh.

I think I may just hang it up and start over next Tuesday. If anybody has an idea, please post it. I'm sure once I get the answer I'll be slapping myself and Duh'ing, but at this point, my brain hurts.

I won't have internet access until I get back next Monday. If I don't reply right away, I'm not ignoring the post. So if you have this one figured out, send it my way please!

I have my answer! I think.

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006170.asp

Posted 10/17/2005

Welcome to AnswerLewis. Last week I asked a question. This week I think I have the answer.

I realized after posting the question that I hadn't really given all of the requirements. So, let me start there.

This was my initial requirement:

I need the difference of the average between the count of transactions in two time periods. Let's try that again. As an example, I need the difference between the average number of transactions in the last 7 days and the last 6 months (including the last 7 days).

After thinking about it, let me say it this way:

I need the difference of the average between the count of transactions in two time periods. Let's try that again. As an example, I need the difference between the average number of transactions in the last 7 days and the last 6 months (including the last 7 days), I will need the difference between the *average number of transactions in the last 30 days and the last 6 months (including the last 30 days), I will need the difference between the average summed amount of the transactions in the last 7 days and the last 6 months (including the last 7 days) and I will need the difference between the average summed amount of the transactions in the last 30 days and the last 6 months (including the last 30 days).*

There are multiple groups (in the millions) in the tables and this query will need to distinguish between groups.

The query needs to be performant, as it will be running against several joined tables. The larger table will have about 1 billion rows. One of the join tables has 50 million rows. In the real tables, the group ID comes from a groups table. It is not part of the transactions table

Ok. So that describes what I am doing a little bit better. If you notice, that description means I will need to add an amount column to my table. I still think for simplicity of the example, I can do this with a single table. So, let's see what our new table looks like:

```
DROP TABLE TRANSACTIONS
/

CREATE TABLE TRANSACTIONS
(
  GROUP_ID  NUMBER,
  TRAN_DT   DATE,
  AMOUNT    NUMBER
)
/
```

You can really insert any random data in these tables that you would like but these are the values I used:

```
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/15/2005 09:29:46 PM', 'MM/DD/YYYY HH:MI:SS AM'), 7.3);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/16/2005 09:29:46 PM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/17/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 21.9);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/18/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 29.2);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/19/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 36.5);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/20/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 43.8);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/21/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 51.1);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/22/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 65.7);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/23/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 58.4);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/06/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 554.8);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/07/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 43.8);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/08/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 36.5);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/09/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 29.2);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/10/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 29.2);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/11/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 21.9);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/12/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/13/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/07/2005 10:13:58 PM', 'MM/DD/YYYY HH:MI:SS AM'), 7.3);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/07/2005 10:14:00 PM', 'MM/DD/YYYY HH:MI:SS AM'), 7.3);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/16/2005 10:14:01 PM', 'MM/DD/YYYY HH:MI:SS AM'), 21.9);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/17/2005 10:14:01 PM', 'MM/DD/YYYY HH:MI:SS AM'), 328.5);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/14/2005 12:00:00 AM', 'MM/DD/YYYY HH:MI:SS AM'), 7.3);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/15/2005 12:00:00 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '04/15/2005 09:29:46 PM', 'MM/DD/YYYY HH:MI:SS AM'), 7.3);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '04/16/2005 09:29:46 PM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '04/17/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 21.9);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '04/18/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 29.2);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '04/19/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 36.5);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '04/20/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 43.8);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '04/21/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 51.1);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '04/22/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 65.7);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '04/23/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 58.4);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '10/06/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 554.8);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '10/07/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 43.8);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '10/08/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 36.5);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '10/09/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 29.2);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '10/10/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 29.2);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '10/11/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 21.9);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '10/12/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '10/13/2005 09:29:47 PM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '10/07/2005 10:13:58 PM', 'MM/DD/YYYY HH:MI:SS AM'), 7.3);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '10/07/2005 10:14:00 PM', 'MM/DD/YYYY HH:MI:SS AM'), 7.3);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '10/16/2005 10:14:01 PM', 'MM/DD/YYYY HH:MI:SS AM'), 21.9);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '10/17/2005 10:14:01 PM', 'MM/DD/YYYY HH:MI:SS AM'), 328.5);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '10/14/2005 12:00:00 AM', 'MM/DD/YYYY HH:MI:SS AM'), 7.3);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
2,  TO_Date( '10/15/2005 12:00:00 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/19/2005 09:28:50 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/20/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/21/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/22/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/23/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/24/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/25/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/26/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '04/27/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/10/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/11/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/12/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/13/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/14/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/15/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/16/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
INSERT INTO TRANSACTIONS ( GROUP_ID, TRAN_DT, AMOUNT ) VALUES (
1,  TO_Date( '10/17/2005 09:28:51 AM', 'MM/DD/YYYY HH:MI:SS AM'), 14.6);
commit;
```

Originally, I said that I had a query using rownum that was a kludge. The reason I said it was a kludge was because it could only return a single row at a time. That would mean re-executing the query for every group. That would be impossible in the real world. Other than that, besides returning multiple rows for each group, the query returned what I was looking for.

This is the base query that did work for me (modified to handle the new requirements):

```
SELECT  group_id,
   tran_dt,
   COUNT(*)  -- First it's the 7 day count
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 7 PRECEDING
                     and 0 FOLLOWING ) count7,
   COUNT(*)  -- First it's the 30 day count
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 30 PRECEDING
                     and 0 FOLLOWING ) count30,
   COUNT(*) -- next the 180 count
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 180 PRECEDING
                     and 0 FOLLOWING) count180,
   SUM(amount)  -- First it's the 7 day sum
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 7 PRECEDING
                     and 0 FOLLOWING ) sum7,
   SUM(amount)  -- First it's the 30 day sum
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 30 PRECEDING
                     and 0 FOLLOWING ) sum30,
   SUM(amount) -- next the 180 sum
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 180 PRECEDING
                       and 0 FOLLOWING) sum180
  FROM transactions dt
  WHERE tran_dt -- getting just last 6 months of data
        BETWEEN trunc(add_months(sysdate, -6))
            AND sysdate
/
```

I'll take this apart piece by piece:

```
FROM transactions dt
  WHERE tran_dt -- getting just last 6 months of data
        BETWEEN trunc(add_months(sysdate, -6))
            AND sysdate
```

This is the same where clause I had before. I removed the trunc around sysdate to get up to the minute transactions. It sets my overall window of data to the last 6 months.

```
   COUNT(*)  -- First it's the 7 day count
```

```
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 7 PRECEDING
                     and 0 FOLLOWING ) count7,
```

- This retrieves a count of all transactions: COUNT(*) -- First it's the 7 day count
- This is an analytic: Over ()
- Grouped by group_id: PARTITION BY group_id
- In ascending ORDER BY tran_dt: ORDER BY tran_dt
- For the set of data in the last 7 days: RANGE BETWEEN 7 PRECEDING and 0 FOLLOWING

I do the same thing for 30 days and 180 days. The SUM(AMOUNT) is basically the same, just a different function, i.e. SUM vs COUNT.

If you run this query, you find that again, the last record in each GROUP_ID group, is in fact the record I need. However, if I just create an outer select and grab rownum = 1, I will only get a single group. That's when my mind clicked on the row_number() function.

ROW_NUMBER() sequentially orders rows IN THE ORDER THEY ARE GROUPED. Sweet. Minor change to the query:

```
SELECT  group_id,
   tran_dt,
   ROW_NUMBER()
     OVER( PARTITION BY group_id
           ORDER BY tran_dt DESC ) rn,
   COUNT(*)  -- First it's the 7 day count
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 7 PRECEDING
                     and 0 FOLLOWING ) count7,
   COUNT(*)  -- First it's the 30 day count
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 30 PRECEDING
                     and 0 FOLLOWING ) count30,
   COUNT(*) -- next the 180 count
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 180 PRECEDING
                     and 0 FOLLOWING) count180,
   SUM(amount)  -- First it's the 7 day sum
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 7 PRECEDING
                     and 0 FOLLOWING ) sum7,
   SUM(amount)  -- First it's the 30 day sum
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 30 PRECEDING
                     and 0 FOLLOWING ) sum30,
   SUM(amount) -- next the 180 sum
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 180 PRECEDING
                       and 0 FOLLOWING) sum180
  FROM transactions dt
  WHERE tran_dt -- getting just last 6 months of data
        BETWEEN trunc(add_months(sysdate, -6))
            AND sysdate
/
```

I added:

```
   row_number()
     OVER( PARTITION BY group_id
           ORDER BY tran_dt DESC ) rn,
```

This would give me a sequentially ordered number for each record by GROUP_ID in TRAN_DT order. Therefore, I would have a number 1 for each group and that number 1 would be associated with the latest TRAN_DT. That latest TRAN_DT is the record I'm looking for.

Now, I can wrap all of this up into my final query by adding an outer select:

```
SELECT group_id,
       (count7 / 7) - (count180/180) avg180_7,
       (count30 / 30) - (count180/180) avg180_30,
       (sum7 / 7) - (sum180/180) sum180_7,
       (sum30 / 30) - (sum180/180) sum180_30
FROM (
SELECT  group_id,
   tran_dt,
   ROW_NUMBER()
     OVER( PARTITION BY group_id
           ORDER BY tran_dt DESC ) rn,
   COUNT(*)  -- First it's the 7 day count
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 7 PRECEDING
                     and 0 FOLLOWING ) count7,
   COUNT(*)  -- First it's the 30 day count
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 30 PRECEDING
                     and 0 FOLLOWING ) count30,
   COUNT(*) -- next the 180 count
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 180 PRECEDING
                     and 0 FOLLOWING) count180,
   SUM(amount)  -- First it's the 7 day sum
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 7 PRECEDING
                     and 0 FOLLOWING ) sum7,
   SUM(amount)  -- First it's the 30 day sum
```

```
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 30 PRECEDING
                     and 0 FOLLOWING ) sum30,
   SUM(amount) -- next the 180 sum
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 180 PRECEDING
                       and 0 FOLLOWING) sum180
  FROM transactions dt
  WHERE tran_dt -- getting just last 6 months of data
        BETWEEN trunc(add_months(sysdate, -6))
            AND sysdate
)
WHERE rn = 1
/
```

The outer select only contains my select list (which defines my differences and averages):

```
SELECT group_id,
       (count7 / 7) - (count180/180) avg180_7,
       (count30 / 30) - (count180/180) avg180_30,
       (sum7 / 7) - (sum180/180) sum180_7,
       (sum30 / 30) - (sum180/180) sum180_30
FROM (
```

And the where clause:

```
WHERE rn = 1
```

This where clause acts like a WHERE ROWNUM = 1 but returns a record for each group_id. My final result set, based on the data I showed above, was:

```
  GROUP_ID   AVG180_7  AVG180_30   SUM180_7  SUM180_30
---------- ---------- ---------- ---------- ----------
         1 1.95952381 .516666667 27.4387302 23.6844444
         2 .905555556 .338888889 12.0507937 21.0888889
```

And that seems to meet my requirements. If you do see anything that I'm missing or if something looks wrong, please let me know.

Can I say this is absolutely the best answer possible? No. It will be tested in the real world and if it's lacking, I will go back to the drawing board. But it does what I need it to do in a single pass through the data. And because of the volume of data I have, that was an important criterion. It also does these things in a single statement. That statement is easily maintainable and if I have another 6-month stat requirement in the future, I can easily add it. And that's important to me also.

Man, I love this stuff! As a wise man once said, "Analytics Rock! Anlytics Roll!"

BTW, I enjoyed my vacation. My wife and I had a great time at the St Pete pier. It has a nice little aquarium and some restaurants. The view is amazing. We also went to Haslan's, Florida's largest used bookstore. I could spend days there. They had a copy of Kimball's data warehouse lifecycle toolkit for $17.

My son got sick at the in-laws and my wife picked up a bug also. They're both relaxing on the sofa and I'm in here writing queries. And attending to their needs of course. heh

Correction to my Analytic Answer

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006282.asp

Posted 10/24/2005

So close and yet so far. On October 13, I posted a question. I followed that up with an answer. Turns out, that answer was incomplete.

Instead of returning the difference between the average transaction count in the last 7 days and average transaction count in the last 6 months, my query was returning the count of transactions for 7 days prior to the date of the last transaction. Not quite what I intended.

As an example, let's say that today's date/time is 10-15-2005 10:00:00. I have the following data:

```
Tran_Dt                Amount
10-12-2005 09:00:00      10
10-10-2005 08:45:00      15
10-08-2005 07:30:00      20
10-06-2005 06:00:00      25
10-05-2005 12:30:00      30
10-04-2005 12:30:00      30
```

If I ran my query from 10-15-2005, I would expect to get a count(*) and sum(amount) for the last 7 days as count = 2 and sum = 25. Instead, the way my query was written it was starting at the date of the last transaction and going back 7 days so the actual results were, count = 5, sum = 95. Definitely not what was intended. The reason I didn't see it earlier is because I was using fake data.

So let's take a look at the original query and see how we can fix it. I'm stripping this down to a single return value and I'm dropping the outer select for brevity. At the end, I'll put the full select statement back together.

```
SELECT  group_id,
   tran_dt,
   ROW_NUMBER()
     OVER( PARTITION BY group_id
           ORDER BY tran_dt DESC ) rn,
   COUNT(*)  -- First it's the 7 day count
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 7 PRECEDING
                     and 0 FOLLOWING ) count7,
   SUM(amount) -- next the 180 sum
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN 180 PRECEDING
                       and 0 FOLLOWING) sum180
```

```
  FROM transactions dt
  WHERE tran_dt -- getting just last 6 months of data
        BETWEEN trunc(add_months(sysdate, -6))
            AND sysdate
/
```

If we refer back to my original answer we can see that the key piece giving me, what I was thinking then, the last 7 days, was:

```
RANGE BETWEEN 7 PRECEDING and 0 FOLLOWING
```

BTW, what I'm saying here also applied to the previous 30 days or 180 days.

The problem with this is that I am always starting with the last record (that's why I had the row_number() function). So, what if the last transaction was 3 days ago? I want to always start with right now as my base point.

This threw me for a loop for a while. But then I decided to try something that I wasn't sure if I could even do. I replaced the 7 in the above statement with a case construct:

```
           RANGE BETWEEN
              CASE
              WHEN tran_dt < sysdate - 7 THEN
                0
              WHEN tran_dt BETWEEN sysdate - 7 and sysdate - 6 THEN
                1
              WHEN tran_dt BETWEEN sysdate - 6 and sysdate - 5 THEN
                2
              WHEN tran_dt BETWEEN sysdate - 5 and sysdate - 4 THEN
                3
              WHEN tran_dt BETWEEN sysdate - 4 and sysdate - 3 THEN
                4
              WHEN tran_dt BETWEEN sysdate - 3 and sysdate - 2 THEN
                5
              WHEN tran_dt BETWEEN sysdate - 2 and sysdate - 1 THEN
                6
              WHEN tran_dt BETWEEN sysdate - 1 and sysdate THEN
                7
              END PRECEDING AND 0 FOLLOWING
```

What this says is that if the tran_dt is older than 7 days, drop it. If it's between 6 and 7 days old, only look back a single day. If it's between 5 and 6, look back 2 days, etc. I do this until I hit the current day and am back to a value of 7.

How cool is that! Man, I was beaming. This was awesome. Then I thought about my 180-day and my 30-day queries. Blech. I would need some HUGE case statements. Moreover, I would need to do this anytime I added a new time period. If the users wanted, say a 45-day time period, I would have a case statement with 45 conditions, 46 to handle greater the greater than condition (and really, 47 to handle future dates although I don't have any in my case, I should still have the else).

So, what do I do with my awesome solution? Like any problem, now that I know it can be solved with the available tools, I break it down and come up with alternatives.

I have a date, tran_dt. I also have SYSDATE. I need to know how many days separate them. So now, I have "**(SYSDATE - tran_dt)**"; I'll call this date_diff. I have a max time frame (in this case 7 days but could be 30, 180 or any other amount of time) and I need to know if the date_diff is less than, equal to or greater than max_time. So now, I have "**7 - (SYSDATE - tran_dt)**"; I'll call this date_cnt. Sticking to my example dates above, this is how it would work out:

If sysdate = 10/15/2005 10:00:00:

```
Tran_Dt                Amount  Date_Cnt
10-12-2005 09:00:00      10      3.958
10-10-2005 08:45:00      15      1.948
10-08-2005 07:30:00      20      -0.104
10-06-2005 06:00:00      25      -2.167
10-05-2005 12:30:00      30      -2.896
10-04-2005 12:30:00      30      -3.896
```

I don't want partial days so I use the ceil function "**CEIL(7 - (SYSDATE - tran_dt))**"; and I get:

If sysdate = 10/15/2005 10:00:00:

```
Tran_Dt                Amount  Date_Cnt
10-12-2005 09:00:00      10      4
10-10-2005 08:45:00      15      2
10-08-2005 07:30:00      20      0
10-06-2005 06:00:00      25      -2
10-05-2005 12:30:00      30      -2
10-04-2005 12:30:00      30      -3
```

That will give me:

```
RANGE BETWEEN
   CEIL(7 - (SYSDATE - tran_dt)) PRECEDING
  and
   0 FOLLOWING
```

But, I'm still not quite there. The negative numbers will be a problem as the PRECEDING expression. I need those to go away. I can do that with the GREATEST function, i.e. "**GREATEST(CEIL(7 - (SYSDATE - tran_dt)), 0)** and that will give me:

If sysdate = 10/15/2005 10:00:00:

```
Tran_Dt                Amount  Date_Cnt
10-12-2005 09:00:00      10      4
10-10-2005 08:45:00      15      2
10-08-2005 07:30:00      20      0
10-06-2005 06:00:00      25      0
10-05-2005 12:30:00      30      0
10-04-2005 12:30:00      30      0
```

And voila, that is exactly what I needed. The final row of the query will still be the one I want.

Unfortunately, the analytic COUNT(*) will always return at least 1 row count if there are any rows at all in the where clause. To get rid of this bogus 1, I had to include a case construct to substract 1 record from the result set, i.e.:

```
        - case greatest(ceil(7 - (sysdate - tran_dt)), 0)
          when 0 then 1  -- If the count should be zero, subract 1
          else 0         -- else, don't subtract anything
          end
```

So here is the final query. You can run this against the set of data I defined in my original answer to test it. I'm leaving my original, hard coded PRECEDING expressions so that you can compare values.

```
SELECT group_id,
       (count7 / 7) - (count180/180) avg180_7,
       (count7_old_way / 7) - (count180_old_way / 180) avg180_7_old_way
FROM (
SELECT  group_id,
   tran_dt,
   ROW_NUMBER()
     OVER( PARTITION BY group_id
           ORDER BY tran_dt DESC ) rn,
   COUNT(tran_dt)  -- First it's the 7 day count
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN
              greatest(ceil(7 - (sysdate - tran_dt)), 0) PRECEDING
              AND 0 FOLLOWING )
        - case greatest(ceil(7 - (sysdate - tran_dt)), 0)
          when 0 then 1  -- If the count should be zero, subract 1
          else 0         -- else, don't subtract anything
          end count7,
   COUNT(*)  -- First it's the 7 day count
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN
              7 PRECEDING
              AND 0 FOLLOWING ) count7_old_way,
   COUNT(*) -- next the 180 count
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN
              greatest(ceil(180 - (sysdate - tran_dt)), 0) PRECEDING
              AND 0 FOLLOWING )
     - case greatest(ceil(180 - (sysdate - tran_dt)), 0)
       when 0 then 1
       else 0
       end count180,
   COUNT(*) -- next the 180 count
     OVER( PARTITION BY group_id
           ORDER BY tran_dt
           RANGE BETWEEN
              180 PRECEDING
              AND 0 FOLLOWING ) count180_old_way
  FROM transactions dt
  WHERE tran_dt -- getting just last 6 months of data
        BETWEEN trunc(add_months(sysdate, -6))
            AND sysdate
)
```

```
WHERE rn = 1
/
```

This has taken a little while to get right but I think I have a robust solution. If we need to add more date ranges in the future, it should be fairly easy to do so.

Highway 11, Slidell, LA – December 23rd, 2005

Chapter 3 - Oracle

Someday I hope to write a book where the royalties will pay for the copies I give away. - Clarence Darrow

Learn Oracle - What is Oracle?

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/005454.asp

Posted 8/22/2005

I guess the first question is really, "What is a database"? A database is an organized collection of data. The data can be textual, like order or inventory data, or it can be pictures, programs or anything else that can be stored on a computer in binary form.

A relational database stores the data in the form of tables and columns. A table is the category of data, like Employee, and the columns are information about the category, like name or address.

Some databases have minimal feature sets and only store data, while others include programming languages, facilities and utilities to support enterprise-level applications like ERP and data warehousing. Oracle is the #1 database and has the most advanced feature set.

Oracle is made up of a set of processes running in your operating system. These processes manage how data is stored and how it is accessed. I will cover these processes in detail in the future; but for now we just need to understand that Oracle is a program that is running in the background, maintaining your data for you and figuring out where it should go on your hard drive.

In almost all relational databases, data is accessed through SQL, or Structured Query Language, and Oracle is no exception. SQL allows you to SELECT your data, INSERT new records, UPDATE existing records and DELETE records you want to get rid of. SQL can be embedded in other languages or you can run scripts of SQL directly against the database.

PL/SQL is the procedural language extension to SQL. PL/SQL is a programming language like C, Java or Pascal. In the Oracle world, there is no better way to access your data from inside a program. SQL can be natively embedded in PL/SQL programs. I will be using both SQL and PL/SQL very heavily in my future articles.

PL/SQL is a feature-rich language geared toward developing database applications. PL/SQL is the procedural language of the database, but it is also the procedural language for most of Oracle's tools. Programs that run inside the database are called stored procedures. These stored procedures are almost always PL/SQL, but can be written in Java.

Some of Oracle's tools to access the database and create programs are:

SQL*Plus has a command line interface. With it, you can access the database and write stored procedures, you can run SQL commands to retrieve data and you can run scripts of either SQL, PL/SQL or built-in SQL*Plus commands, or a mixture of those three things.

Oracle Developer is a 4GL GUI application Builder. With Developer, you can create forms, reports, and graphics. Oracle*Forms and Oracle*Reports are two components of Oracle Developer. Earlier versions created client-server applications, but the more recent versions create web applications that run under the Oracle Application Server (OAS). OAS is a web-based application server sold by Oracle. OAS is licensed separately and is very expensive (as are its closed source competitors). The current version is OAS 10g.

HTML DB is a fairly new application builder geared toward web development (added to the DB with release 9iR2). HTML DB does not need an application server. This tool runs from the database and can be presented to the web using the Apache web server that comes with the database. Since this is not an additional license, it provides a cheaper way to develop applications. Developer is a feature-rich thick client with all of the normal GUI widgets. HTML DB is HTML-based and is very thin and limited to the HTML provided widgets.

Oracle Enterprise Manager (OEM) is the Enterprise GUI tool to manage the database. From this tool, you can perform any action in the database that is required. This tool is primarily used for administration but can also be provided to developers for application tuning and monitoring. In Oracle 10g, OEM also provides Grid control.

There are also a lot of third party tools for accessing the database. For our purposes, our primary tool for data access will be SQL*Plus. In the future, I will cover creating applications with HTML DB and accessing your data with some choice third-party tools.

How about Java? Java is the current hot language, especially on the web. I will also be covering Java in the database in future articles. Java is natively supported by Oracle. Technically, you could use Java to create your stored procedures if you chose, but I am not a proponent of that. When you are working inside the database, I believe that you should only resort to Java when you cannot accomplish a task in PL/SQL. As you'll see, there is very little you cannot accomplish with PL/SQL.

It will be hard to learn Oracle if you can't play with it. So where can you get access to this software? Oracle has a technical web site called OTN (Oracle Tech Net) that provides access to all of Oracle's software and all of the documentation for that software. There are also forums and many other tools to use to help you learn Oracle. You have to register to access it, but registration is free. Go to http://otn.oracle.com/ and get an ID today. In future articles, I will be spending a lot of time accessing this web site to download software and documentation. My next article will cover downloading Oracle 10g and installing it.

As a side note, Oracle provides all of their software with a developer's license. This license allows you to try out the software free of charge. You may not create production applications without a paid license, but development with the tools is acceptable. I am not a lawyer, so I recommend that you review the license when you download any software from OTN.

Also, a note about versioning in Oracle; all products released by Oracle have a version. The current version of the database is 10g Release 1, also written as 10gR1. In addition to this semantic release name, each release has a numeric version. The numeric release for 10gR1 is 10.0.1.3.

Note: This article was previously published in April 2005 on the Suite101.com "Oracle and More!" topic at http://www.suite101.com/welcome.cfm/oracle. This article may have been modified for formatting.

Introduction to Basic SQL, Part 1

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004917.asp

Posted 8/24/2005

SQL, pronounced SEQUEL, is the standard language to access relational databases. SQL is an abbreviation for Structured Query Language. Don't let the fact that it's a language scare you. Using its basic functionality is very simple.

Even though SQL is a standard, every vendor has his or her own extensions and limitations. Oracle is no exception. Until Oracle 9i, Oracle did not support standard join syntax that was supported by databases as simple as MS-Access. However, Oracle has always provided extensions that made it a very robust platform for everything from general user access to ad-hoc queries to high performance real-time applications to data warehousing.

In this article, I will be covering basic SQL. The queries in this article will run against most any Oracle database. The objects that we are querying against differ between database vendors. I will follow this article shortly with an article describing some more advanced SQL functionality and how that functionality differs between Oracle, MySQL and PostgreSQL.

So on to the show! If you don't have a database installed, I recommend that you refer to:
http://www.suite101.com/article.cfm/oracle/115311.

Once you have a database installed, Oracle provides a set of tables and views (don't worry about the difference at this point) called the data dictionary. This data dictionary describes all of the objects in your database.

We will start with a view called ALL_TABLES. ALL_TABLES is a list of all of the tables in the database that you have access. The values that you select will be different from the values that I will select.

The most basic syntax of SQL starts with a SELECT clause and a FROM clause. A SELECT clause tells the database WHAT you want to select. A FROM clause tells the database where to find that data.

Run SQL*Plus (described in the article I mentioned above), login as whatever user you have available and enter:

```
SELECT table_name
  FROM all_tables;
```

When I run my query, I get a list of tables followed by the text "113 rows selected.".

The SELECT clause is TABLE_NAME and the FROM clause is all_tables. The semi colon at the end tells Oracle that you have completed your command. You MUST enter a semi-colon or a slash before Oracle will process you command.

This was a very useful command but it could be made more useful. The list of tables was just that, a list. If we were looking for a specific table, it would be hard with an unordered list like that. Let's try putting some order to it.

Enter:

```
SELECT table_name
  FROM all_tables
  ORDER BY table_name
/
```

There are two differences here.

- There is no semi colon but there is a slash, "/". The slash also tells Oracle that you have completed a command. The slash must appear ion column 1 on a line by itself. A semi colon can follow anywhere as long as there are no blank lines between it and the command.

  Type slash on a line by itself and Oracle will re-execute the previous command. Try it now.

- We also added an ORDER BY clause to our command. The ORDER BY tells Oracle to, oddly enough, order the result set. What did it order by? TABLE_NAME.

Now that is a lot more useful. You can scroll through the list and find a specific table. Scroll through it now and find the table, DUAL. If you don't see this table, something is either not configured properly in your database or not configured properly with your user. DUAL is a special table provide by Oracle. It should always have one row and only have one column. That column, DUMMY, should always have the value 'X'.

Enter:

```
SELECT table_name
  FROM all_tables
  WHERE table_name = 'DUAL'
/
```

You should get only a single row back. If you get more than a single row, you must have multiple DUAL tables in your database. Let's add a new column to our select:

```
SELECT table_name
  FROM all_tables
  WHERE table_name = 'DUAL'
  AND OWNER = 'SYS'
/
```

SYS is a very special user. SYS own most of the underlying objects in a database.

As you can see the WHERE clause limits the data you select. You will hear/read it referred to as the where clause, limit line, restriction line, etc.

So how did I know that the ALL_TABLES view had the column OWNER in it.

Enter:

```
DESC DUAL
```

The DESC command is a SQL*Plus command, not a SQL command. SQL*Plus and SQL command are not case sensitive. That is you may enter them as either upper case, lower case or any combination of case. Text in quotes, called literals, is case sensitive. Try entering:

```
SELECT table_name
  FROM all_tables
  WHERE table_name = 'Dual'
/
```

You should get no rows. ALL_TABLES, and most other data dictionary views, always store data in upper case. There is a way to create objects in lower and mixed case but I consider that such a bad practice that I don't want to be the one to show you how. ;-)

SQL provides useful functions that you can execute in your selects to modify data. Suppose that we wanted to find the third character of the table name for each table. We would enter:

```
SELECT table_name, substr(table_name, 3, 1)
  FROM all_tables
/
```

You can see that there are two columns in you return list. SUBSTR is a SQL function that means sub-string. Basically, it returns a subset of the value in a string.

If you wanted to see if a value is in a string you could enter:

```
SELECT table_name, instr(table_name, 'A')
  FROM all_tables
/
```

INSTR is a function that finds the location of a substring in a string. The above function call says return the location of the substring A for the column table_name from the table all_tables. If the new column is a 0, that table name does not contain an A. If it is a number higher than 0, than is the first location in the table name of the character A.

And that is basic SQL in a nutshell. All of this syntax will work in most any database but only Oracle has the DUAL table. Data dictionaries also vary between databases. In a sneaky way, I have also introduced you to more advanced Oracle concepts like the DUAL table and a data dictionary.

Today we covered the SELECT, FROM, WHERE and ORDER BY clauses of SQL. The SELECT is the WHAT, FROM is the SOURCE, WHERE is the LIMIT and ORDER BY is the ORDER OF. With these four clauses, you can access just about any relational database in existence.

Note: This article was originally published in May 2005 on the Suite101.com "Oracle and More!" topic at http://www.suite101.com/welcome.cfm/oracle. This article may was modified from the original.

Oracle 10g - Nifty little features

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004917.asp

Posted 7/11/2005

There have been a lot of blogging about the new Oracle features like SQL Tuning Advisor, SQL Access Advisor, AWM, the scheduler, etc. Here are a few items that I haven't heard much about but are nifty little time savers. These are the kinds of things that make life just a little easier.

1. glogin.sql and login.sql are now executed after every connection, not just when SQL*Plus is started as in prior versions. If you use CONNECT to reconnect to the database, these files are re-executed.

2. SQLPLROMPT - Now provides parameters so you can get rid of that kludgey sql in your login.sql. Just use this to get USER@DB.

```
set sqlprompt '&_user.@&_connect_identifier.>'
```

If JOHN_DOE is connected to DB1, the syntax on the line above will give you

```
JOHN.DOE@DB1>
```

as a sql prompt. Put this in your login.sql and every time you connect it will give you a prompt that says who and where you are.

3. Oracle JDBC thin client driver now supports Oracle collections. This allows you to pass collections back and forth between Java and PL/SQL.

4. Regular Expression support including using regular expressions in where clauses. If you have never used regular expressions, this won't impress you. If you have used regular expressions, you're probably wondering why it took so long.

5. UTL_COMPRESS - compress your data using zip. Serve up some compressed data for those low bandwidth days.

6. PL/SQL is now an optimizing compiler. It will re-write your code if it thinks it can do it better. It will automatically re-write a cursor for loop to use BULK processing.

7. SQL now supports case insenstive queries. No more UPPER(x) = UPPER(y). Use

```
alter session set nls_comp=ansi;
alter session set nls_sort=binary_ci;
```

and then do your query.

8. Synonym creation now supports 'CREATE OR REPLACE SYNONYM' syntax. Public synonyms also support this syntax. Using create or replace will not invalidate views or procedures. NOTE: Reader Martin Strobl pointed out that this feature has really existed since 9ir2.

I think this one is worth a little sample:

```
jblow@db1>create table abc ( text1 varchar2(10) );

Table created.

jblow@db1>create or replace synonym def for abc;

Synonym created.

jblow@db1>create or replace procedure xyz as
2           v_dummy varchar2(10);
3         begin
4           select text1 into v_dummy from def;
5         end;
6 /

Procedure created.

jblow@db1>select status from user_objects where object_name = 'XYZ';

STATUS
```

```
-------
VALID

jblow@db1>create table ghi ( text1 varchar2(10) );

Table created.

jblow@db1>create or replace synonym def for ghi;

Synonym created.

jblow@db1>select status from user_objects where object_name = 'XYZ';

STATUS
-------
VALID
```

Nice. I have places where I do this now by dropping and recreating which means I have some downtime. It's minimal but with the new way there will be none.

I believe all of these nifty little features were added with 10gR1. I had a little downtime this weekend to do some reading. There are so many new features in 10gR1 that it's hard to keep up. I haven't even had time to begin looking at 10gr2.

Oracle 10g vs PostgreSQL 8 vs MySQL 5

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004209.asp

Posted 8/22/2005

This is my comparison of installing and getting started with Oracle 10g, PostgreSQL 8 and MySQL 5. This is what I consider the comparison of state of the art for three categories of DB: Commercial vs. Academe vs. Internet Model. This is a comparison from the view of a new user wanting to install a database to learn. I'm trying to keep the viewpoint of a home or small business user.

Full Disclosure: I am strongly biased towards Oracle and fully expected no real competition.

Basic Criteria

License: I will not get into a debate over open source and closed source. Oracle is a commercial database and requires licensing to use. MySQL and PostgreSQL are both open source projects. Oracle provides a free developers license for you to "test out" and create prototypes.

Configuration: I am installing on an older machine with 256 Megs of Ram and a 40 Gig hard drive. The CPU is a 633 Mhz Celeron. The OS is Windows 2000 Professional. I want to install at the low end of the spectrum to gauge the ability for home user type installations.

OS: I chose to perform this comparison under Windows, as I believe that that is still the most common platform that new people to the world of databases will use. By that, I mean people installing at home. For new corporate users, the odds are good that they will not have a choice of either OS or database so a comparison is pointless.

Versions:

PostgreSQL 8.0 - PostgreSQL 8.0 is a very new product. I chose 8.0 as it is the latest version and it natively supports Windows. This is the first version that actually does support Windows without an emulator or third-party recompilation.

MySQL - MySQL 5.0 is also a very new product. As a matter of a fact, the version I used, v5.0.4, is actually a beta version. I argued with myself about whether I should use the stable v4 or the beta v5. I chose v5 because v4 does not compete with PostgreSQL or Oracle. V4 does not have stored procedures, triggers, views, etc. Without those features, I would not consider it as a contender. The upside to that is that v5 does support those features.

Oracle 10g - This is the latest version of Oracle and was released last year. In my opinion, Oracle 10g is the gold standard of databases. Nevertheless, I also recognize that it is very expensive and requires quite a bit of knowledge to support in a production environment.

I am ranking this comparison on six factors:

Documentation and Getting Started Support
Ease of installation
Ease of Verifying Successful Installation
Creation of Non-Admin User
Time to Run First Query
Resource Requirements

Ranking Values: I rank each factor a score of 1 through 10, with 10 being the best or highest.

Documentation and Getting Started Support

PostgreSQL - I found a lot of support for previous version (under Linux, Cygwin, etc) but almost nothing for 8.0 under windows. The PostgreSQL documentation website, http://www.postgresql.org/docs/, provides very good, very complete documentation. In my opinion, one of PostgreSQL's strengths in the past has been the documentation.

That web site includes a very good installation document. However, post installation and getting started documentation is very limited. I hope that that will change in time.

I googled "PostgreSQL getting started" and got thousands, mostly version 7 hits. PostgreSQL has some Usenet newsgroup support but I noticed they were not very active.

PostgreSQL Documentation Ranking: 5

MySQL - Because v5 is so new, there is not very much documentation yet. There is a world of documentation for MySQL v3 and v4. Check Amazon or BN.com. MySQL seems to be the choice for a lot of smaller Internet and java-oriented companies.

Because v5 is beta, I will not rank this factor according to the documentation available for that release. V4 has amazing documentation and I fully expect v5 to have the same by the time the software is release to production. I will rank according to v4 documentation.

I googled "MySQL getting started" and got millions, mostly version 4 hits. I could find no MYSQL specific Usenet newsgroup support but I noticed a some postings for MySQL support on the comp.databases group.

MySQL Documentation Ranking: 8

Oracle - Oracle, as a commercial product, has a large staff of full time technical writers. Every feature is spelled out and many features receive their own books. The installation documentation provided by Oracle is superior and where that is lacking it is made up by third party sites dedicated to Oracle. The documentation, as well as the documentation portal is very mature.

In addition to installation guides for just about every OS, Oracle provides a "2 day DBA" guide. This document is indispensable for new users.

I googled "Oracle getting started", and received millions of hits. Oracle also has robust Usenet news group support.

Oracle Documentation Ranking: 10

Ease of Installation

PostgreSQL

I haven't previously used PostgreSQL outside of Cygwin (a Unix environment for Windows). I was surprised at the feel of the installation. It seemed very native and ran flawlessly. It was also a very quick install.

PostgreSQL installed the quickest. According to the documentation, you can install PostgreSQL without administrative rights. As this is the first version of PostgreSQL to support windows (without Cygwin), I wanted to test the Services support. PostgreSQL installed as a service and came up without problems after rebooting.

PostgreSQL Installation Ranking: 10

MySQL

I ran a web site in the past, mainly for my own amusement, and used MySQL as the database. The technology stack I used was Java/Apache/MySQL. I was fully expecting MySQL to win the installation competition based on experience. However, MySQL was the only install to fail during installation. I performed a little cleanup and then re-ran the install. It failed again. I cleanup again, rebooted and re-ran the installation. This time it installed. I'm not sure what caused the error. It was very cryptic but it was the same both times it failed. I was able to bulldoze past that though. Remember, this is a beta release.

MySQL also installed as a service and on reboot came up with no errors. The install, when it ran correctly, ran quickly.

MySQL Installation Ranking: 5

Oracle

Oracle again proved that it is a very mature product. Of the three, Oracle is the only one that can NOT be installed without admin rights. It would be nice for this to be an option.

It did, however, install easily. I received no errors. Oracle was the only install to point me towards the next step at the end of the install. This is a sure fire method to support new users and requires almost no effort. All databases should provide this kind of support.

Where Oracle really took a hit was in the amount of time it took to install. This is a real issue for home users. There were a couple of spots where I thought the machine might be hung as I received no feedback for an extended amount of time.

Oracle Installation Ranking: 7

Ease of Verifying Successful Installation

To verify the success of the install was a little different but all three provide data access tools. I followed the instructions in the provided documentation and was able to access the server in all three instances.

Verification Ranking for all 3: 10

Creation of Non-Admin User:

PostgreSQL: PostgreSQL uses the standard CREATE USER syntax and I was able to perform this task easily. The documentation walks the user through this task (which is required for security reasons).

PostgreSQL Create User Ranking: 10

MySQL: In v5, MySQL also uses the CREATE USER syntax. Prior releases did not. Do to the fact that the documentation is not yet complete; it is not completely obvious that CREATE USER is available. For a new user, this can be confusing.

MySQL Create User Ranking: 7

Oracle: Oracle uses the standard CREATE USER syntax and I was able to perform this task easily. The documentation walks the user through this task.

Oracle Create User Ranking: 10

Time to Run First Query

All three database documentation sets provide a tutorial for getting connected and running your first query.

All three First Query Ranking: 10

Resource Requirements:

PostgreSQL - I was amazed at PostgreSQL's minimal need for resources. I performed some tests like opening web browsers and MS-Word while connecting remotely and could see no impact at all on my test machine.

I couldn't find the minimum hardware specs but 256Meg on an old machine seems to be more than enough.

PostgreSQL Resource Ranking: 10

MySQL - MySQL's minimal need for resources is well known. That is one of its biggest selling points and always has been. I performed the same test that I preformed for PostgreSQL and MySQL performed as well as PostgreSQL.

Oddly enough, I couldn't find the minimum hardware specs for MySQL either but 256Meg on an old machine seems to be more than enough.

MySQL Resource Ranking: 10

Oracle - Oracle uses the most memory, and Oracle 10g uses even more than previous versions. It really would run better on a 512M machine and I would recommend 1gig.

On the test box, it takes forever (well, many long minutes) for Oracle to even start up. When I opened Word, I started getting swapping and stalling. With multiple remote connections, the database came to a standstill. Oracle will install and run in 256 Megs but I don't recommend it.

Oracle Resource Ranking: 3

So how does each stack up? Here is the ranking in tabular format:

	PostgreSQL	**MySQL**	**Oracle**
Documentation	5	8	10
Installation	10	5	7
Verification	10	10	10
User Creation	10	7	10
First Query	10	10	10
Resources	10	10	3
Total:	55	50	50

So, that is the install and first use ranking. That is not at all what my expectations were. I think it's great that all three rated so high. I was not expecting that or that Oracle and MySQL would tie. I was definitely not expecting PostgreSQL to come in first.

However, installation is only a small piece to using a database. There is language support, administration, additional usability features, security and third-party support. In the near future I will write up a comparison of these components.

Note: This article was previously published in April 2005 on the Suite101.com "Oracle and More!" topic at http://www.suite101.com/welcome.cfm/oracle. This article may have been modified for formatting.

West End, New Orleans, La – December 24th, 2005

Chapter 3 - Open Source

Computations are everywhere, once you begin to look at things in a certain way. - Rudy Rucker

Oracle and the Open Source Database

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004209.asp

Posted 5/19/2005

There has been a lot online recently about open source databases and the possibility of Oracle open sourcing their database. Oracle has "embraced" the open source community by certifying certain products with Oracle and by providing PHP and Linux platform support. With the open source lovefest, some have suggested Oracle should (some even say WILL) go open source. I think it's more appropriate to ask, "Can they go open source?"

In the interests of disclosure let me say I own Oracle stock and I don't belong to the open source religion. I like open source and I'm glad open source is available. I don't believe that open source is the only valid way to create software though.

It's true that Oracle's open source competitors are getting better and that larger enterprises are starting to consider open source alternatives. I've been comparing features in Oracle against those in MySQL and PostgreSQL and I'm finding PostgreSQL to be a very robust RDBMS.

So I got to thinking about the possibility of Oracle going open source and there are three reasons that I ask, Can Oracle go open source?

1. FUD - Fear, Uncertainty and Doubt. Unfortunately, I feel that enterprise support for open source is a fickle beast. It's led mainly by technical folks who see the immediate use and not necessarily the TCO and other impacts. The more financial oriented actually tend towards more conservative approaches and well-entrenched software names. IBM, Oracle and MS sell well because they are known as big name, big dollar companies. They are trusted by other big name, big dollar corporations. Publicly traded is good, open is bad. That's not my opinion but it is the opinion of many.

And those on the fence are waiting for the other shoe to drop. I have to ask, what happens when there is a major open source scare? What if a major flaw is found in a common open source application? That is the FUD part. It will happen and with all of the media coverage of open source it will be on the front-page of every IT magazine and probably on the front page of many business magazines. It will probably be debated on the major news outlets.

Technical people will read it as par for the course. No software is completely bug free. And open source by definition tends to have more eyes checking the code.

Non-technical business people won't care that it's a single bug. The problem will be fixed and the developers will get back to work but the bug will linger in memory. The big names employ marketing to spin bugs into reasons why you shouldn't move towards open source. It's just a fact of business.

All it takes to fail in the eyes of enterprise buyers is a single incident. It doesn't matter that the same kinds of incidents happen to closed proprietary software everyday. It's the appearance of a failure that will have the anti-open source community (big Comerical software) saying, "See we told you so!"

Oracle cannot, the shareholder will not, allow the Oracle corporation to be placed in this FUD environment. If Sun can't figure out how to release Java, which it isn't even selling, to open source, how can anyone expect Oracle to do so.

2. Finances. Oracle makes its money selling database licenses. When it's not making money selling database licenses, it's selling licenses to other stuff.

Checkout Oracle's Investor 2Q05 supplement.

Oracle is in business to make money. Selling databases is making them money. There is no way the share holders will allow Oracle to open source it's number one money maker. Open doesn't mean free but that is how most of the world still sees it. This one seems the most obvious to me. Check out the supplement and tell me if I'm reading it wrong. Shareholders won't allow the database to become open source.

3. Competition. There is no way Oracle will release that kind of intellectual property to its competitors. Everything that Oracle does well would appear in every competitor's product (not stolen code, just ideas). And everything that it does poorly would be hung up to wave in the wind like banners.

Have you seen the ad from IBM that basically says Oracle quit providing support for 8i so IBM must be better? Can you imagine what they would do if they got a hold of some ugly source?

Again, from a marketing standpoint, going open source makes no sense for Oracle. At least not for the database. When I get a chance later this week, I'll post what I think Oracle can open source and what I think Oracle can do to regain the mindshare they've lost because of their licensing practices.

As always, leave me a comment if you have a question, thought or critique!

And a PS. I really like freeware. Kind of the 80s and 90s version of free software. As a young developer, I wrote some freeware utilities and even some shareware (Ever hear of Simple Install? Neither did anyone else, heh). So I like to browse the freeware newsgroups and lurk. Anyway, the other day someone mentioned a piece of software, the HTML Editor Arachnophilia, that I hadn't heard of in a long time. It's been re-written in Java and after playing around with it this weekend, it brings back some memories (I don't do much HTML anymore). Anyway, it has a unique, optional licensing scheme and this discussion seems like a good place to mention it. Check it out.

A Kinder, Gentler Oracle

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004277.asp

Posted 5/20/2005

In my last entry I discussed the fact that I don't think Oracle can even consider open sourcing the database. It doesn't make sense for them to open almost any of their software.

One place where it would make sense is in their application development stack. Oracle's direction is clearly java. Forms is getting less and less attention. That is a product that's begging to be opened up. It could even follow a JCP type process rather than being truly open. There are a lot of companies with heavy investments in Forms that would be willing to maintain and upgrade the product.

Oracle reports is still too integrated with too many other products to think about opening that up.

Without resorting to open source, there are some other things they can do to grow mind share though. Free versions for instance. It seems oxymoronic that free software can add to the bottom line but follow along for a minute.

Quit selling Oracle Standard Edition One and start giving away the Oracle DB Community Edition (ODBCE). How much can they really be making on Standard Edition One? Oracle already provides all of their tools completely free for development purposes. Take that one step further and provide a free database for those companies who don't need to scale out and don't need the functionality of EE. When applications get developed in ODBCE that's still market share for Oracle.

And since these small companies are growing in-house expertise in Oracle, when those applications grow and do need to scale, that's additional licenses that will go to Oracle instead of to MS or an open source database as they would now.

A community edition of IAS would be nice also. Again, smaller companies would then use it and grow their own expertise giving Oracle market share and future revenue that it's losing now. The community edition can drop some of the more enterprise-ish features like HA, Identity management, BI, etc. If a company needs those, they would be required to purchase the full license. Again, giving software away to people who probably wouldn't buy it anyway but having them in the Oracle camp when they are in a position to purchase.

I would also suggest that Oracle lower their prices across the board. It seems to me that the mantra now is make money at all costs. It will hurt them in the long run. All sales people are pushy but Oracle's gets ridiculous. Although they are willing to deal to meet their targets, dropping the per CPU cost (and not upping it for dual core chips) would get more interest from smaller players.

Oracle is getting a lot of media play for integrated java, integrating PHP, supporting and developing on Linux, etc. They get a lot from Apache and other open source initiatives. It's not in their best interest to open source their products but it is in their best interest to clean up the reputation that they are too expensive to be an entry-level database. Eventually they will be considered too expensive for enterprises. Build the support at the bottom and the rest will follow.

But that's just my opinion.

Oracle and the Open Source Competition

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004312.asp

Posted 5/27/2005

I recently wrote about Oracle and any possibilities I feel that Oracle will open source their database. I thought that a good companion to that article is one where I talk a little about the open source competition Oracle faces in the DB arena. So that's what I'm doing now.

I recently had the opportunity to play with, and compare, MySQL 5.0 and PostgreSQL 8.0. Below are some opinions and some light reading if you're interested in the topic.

Oracle, MySQL and PostgreSQL all run on most OSes and all have some kind of support for most languages. Oracle and PostgreSQL both have a built-in procedural language that puts the power of the database close to the database. Code wise, my mantra is, if it can go in the database, it should.

I found a good link comparing the basic features of MySQL and PostgreSQL. This is one of the best articles of its type that I've seen. It's over a year old so it doesn't cover the latest details perfectly, but still a good read for those interested.

Update 5-26-2005. I found a new link comparing several open source databases and thought I should pass it along. I can't say how accurate it is. I've never even read more than a brief description of a couple of these but it was last updated March 2005 so it may be worthwile.

Update 5-27-2005. Oddly enough, two articles just came out in eWeek that are pertinent to this discussion. They are here, Open Database Startup Taps PostgreSQL, and here, At Last, a Worthy Open-Source Alternative to Oracle. Looks like things are heating up.

MySQL - I used MySQL for a while back in the 3.x and 4.x days. It's come a long way since then. MySQL 4.0 was a complete no show for the enterprise but 4.1 was a good step forward. The next generation, MySQL 5.0, is still in beta although it adds some required features for the enterprise. As usual, its performance can't be argued with but it still seems lightweight feature wise to me. Right now, it feels like an Oracle 7 killer not an Oracle 10g killer. Maybe in the future.

Also an annoyance for me, is FUD (Fear, Uncertainty and Doubt) being spread like Oracle requiring 10 CDs to install. Oracle doesn't take 10 CDs to install and this kind of FUD is good for no one.

One article that I read that looks good for the future is that MySQL is hooking up with Business Objects. And, of course, I'm sure you're heard of LAMP. That seems to be a standard for a lot web sites.

So at this point, whether you use a production version (4.1) or a beta version (5.0), MySQL is just not the Oracle killer yet. In the future, who can say? Is there a place for MySQL in the enterprise? My opinion is yes, there are some high volume, lightweight applications that fit MySQL perfectly. Nevertheless, the bottom-line for now: Oracle is innovating and MySQL is playing catch-up.

PostgreSQL - 8.0 is the new production version for PostgreSQL. I played with Postgres way back when; I can't even remember the version. It ran under Cygwin and I never really got to much depth with it.

I have to admit I was impressed recently when I download 8.0 and started playing with it and reading the docs. The feature set is impressive and I think it can run on just about any OS you care to name. Just what's new is a laundry list. The feature list really compares to Oracle.

If you read my recent Sorcerer of OO, Part 1 and Part 2 you know I'm into the OO features of the database. I like the PostgreSQL implementation, at least in theory. I haven't actually implemented anything with it.

I also recently had occasion to check out the datawarehouse appliance, Netezza. It's custom hardware running Linux and PostgreSQL. Although I think it's an older version, maybe 7.x. That's just a guess on my part though.

About the biggest disappointment I have with Postgres is its lack of marketing. The personalities at Oracle and MySQL make sure that both are in the news whenever possible. I just don't hear about Postgres.

Based on mostly reading documentation and a little bit of playing around, I dub PostgreSQL 8.0 the silent killer. It has the features and if it's supporting huge terabyte databases in the Netezza appliance, it has the performance. PostgreSQL needs to grow some personalities to market it and make it known to the world.

PostgreSQL can be a real open source challenger to Oracle, IBM and MS. And that's good for everyone.

Comments

James 5/24/2005

Your thinking is flawed in believing that MySQL is playing catchup. Did you ever ask the question of how many folks actually need all of the features that Oracle provides. I suspect that the current implementation of MySQL and PostGres will support 99.9% of the requirements for 99.9% of all applications within corporate America...

LewisC 5/24/2005

James, Thanks for the feedback.

I would say that Postgres probably could meet 90-95%. I don't believe MySQL can hit that high of a percentage. I'm in an enterprise environment and I have used pretty much everything Oracle provides at one time or another.

I take points away from PostgreSQL for some specific features I use often that they're missing (AQ as an example). I think a lot of those features that are missing could be created using native PostgreSQL built-ins though. Hence the 90-95%.

Every application doesn't require every feature but in a large organization, there are many types of applications. If one application needs a feature, it's a requirement. Can a project not do a particular something and get by? Possibly. But I would rather have a database that can do that something out of the box.

As far as catch-up goes, MySQL 5.0 adds stored procedures (sort of), triggers (sort of) and views. That sounds like catch-up to me. Oracle added these features in v7.0 (1993ish).

Even the MySQL CEO said (in one of the links above) that MySQL hadn't quite caught up to Oracle but would in the future. That's catch-up.

I don't think my thinking is flawed. Of course if my thinking is flawed, it probably doesn't matter what I think. ;-)

But I do think we'll just have to disagree on this one.

LewisC

Josh Berkus 5/24/2005

Lewis,

A few things:

1) We don't have the marketing because PostgreSQL isn't a company. So we're not going to promote ourselves in the same way that MySQL or Oracle do.

2) I work for GreenPlum (www.greenplum.com), the "other" Teradata alternative. It's also PostgreSQL-based.

3) Netezza uses PostgreSQL-based code, but it's heavily hacked.

4) Marten of MySQL has said that MySQL AB is not really that interested in the Oracle market; they're interested more in "new markets". We, on the other hand, have been pulled pretty strongly into the role of Oracle alternative because of user demand. I get e-mails every day from users wanting migration information.

5) My general experience is that 60% of Oracle users can migrate to PostgreSQL for a reasonable cost (defined as much less than the cost of more Oracle licenses). Companies like EnterpriseDB seem to be working to raise that percentage.

6) Oracle users with light performance requirements could also consider Fyracle, a version of Firebird with something like 95% Oracle syntax-compatibility.

7) Oracle, in my opinion, isn't innovating much at all. This is what's made them such a soft target for OSS database systems. We'll pass them in less than 2 years; MySQL probably in a couple more. If you want to see innovation, check out SybaseIQ, StreamBase or any number of OSS projects.

--Josh Berkus
PostgreSQL Project
(opinions stated above are my own personal opinions)

LewisC 5/25/2005

Josh,

Thanks for the great feedback. I'll respond to your points individually.

1) We don't have the marketing because PostgreSQL isn't a company. So we're not going to promote ourselves in the same way that MySQL or Oracle do.

True and a good point but that's why I said personality instead of CEO. I was thinking more along the lines of Linux and Linus. You don't need to be a company for someone to thump the drum and draw some attention. Although, the motivation is greater if you're being paid.

2) I work for GreenPlum (www.greenplum.com), the "other" Teradata alternative. It's also PostgreSQL-based.

I haven't heard about Greenplum although now I know it's an open source database server based on PostgreSQL specifically designed for BI. If you don't mind, I'll contact you offline for some details. I think I might like to set up a prototype of something I've been working on and compare the differences. I might get a blog or two out of the experience. ;-)

3) Netezza uses PostgreSQL-based code, but it's heavily hacked.

How so? What are you basing that information on? Don't get me wrong, I'm not disputing it but I would like to keep it factual with URL references if possible.

4) Marten of MySQL has said that MySQL AB is not really that interested in the Oracle market; they're interested more in "new markets".

I have read that. I don't really believe it but I have read it.

We, on the other hand, have been pulled pretty strongly into the role of Oracle alternative because of user demand. I get e-mails every day from users wanting migration information.

Due to Oracle licensing, I can understand that. Their licensing is exactly why I wrote my two blogs from last week, Oracle and the Open Source Database and A Kinder, Gentler Oracle.

5) My general experience is that 60% of Oracle users can migrate to PostgreSQL for a reasonable cost (defined as much less than the cost of more Oracle licenses). Companies like EnterpriseDB seem to be working to raise that percentage.

I will definitely have to defer to you on this one. Thanks for the stats. I thought it would be higher but that opinion was based on very preliminary information.

6) Oracle users with light performance requirements could also consider Fyracle, a version of Firebird with something like 95% Oracle syntax-compatibility.

I've been wanting to play with firebird but just haven't had the time. I have heard very good things about it. I haven't heard of Fyracle. Interesting name. I'll add that to my to bottomless to do list.

If it has 95% syntax compatibility, what is its feature compatibility?

7) Oracle, in my opinion, isn't innovating much at all. This is what's made them such a soft target for OSS database systems. We'll pass them in less than 2 years; MySQL probably in a couple more.

Here we disagree. XML DB, Streams, RAC, Grid, ADM, Bitmap Joins, Contexts, etc. The list goes on and on. That's not innovation? I think the only chance open source has to supplant Oracle, is Oracle's own stupid licensing.

And doesn't your statement that you will pass them in two years, and MySQL in a couple more, define the phrase "catch-up"?

If you want to see innovation, check out SybaseIQ, StreamBase or any number of OSS projects.

I completely agree SybaseIQ is a great technology. But it's a rehash of technology invented in the 70s. Timely and very useful, but a rehash. I'm not familiar with StreamBase. I would hope OSS is innovative. That should be its strength. A lot of minds collaborating.

Thanks for the comments. I appreciate the feedback.

LewisC
No Project In Particular

Josh Berkus 5/25/2005

Lewis,

LC: True and a good point but that's why I said personality instead of CEO. I was thinking more along the lines of Linux and Linus. You don't need to be a company for someone to thump the drum and draw some attention.

JB: Linux has been boosted by IBM spending $100 million marketing it. That wasn't Linus' doing that I know of. And we are working on promoting PostgreSQL; Google for us and you'll see a fair amount of press.

LC: I haven't heard about Greenplum although now I know it's an open source database server based on PostgreSQL specifically designed for BI.

JB: GreenPlum's product has 2 parts: DeepGreen PostgreSQL, which is a single-server version of PostgreSQL designed for BI use (currently in alpha), and DeepGreen MPP, which is a multi-server, multi-terabyte solution which is proprietary but based on PostgreSQL.

LC: How so? What are you basing that information on? Don't get me wrong, I'm not disputing it but I would like to keep it factual with URL references if possible.

JB: Well, I know because the president of Netezza Inc. told me so, in person at LWE-SF. If you google, he may have said it to the press as well.

LC: I will definitely have to defer to you on this one. Thanks for the stats. I thought it would be higher but that opinion was based on very preliminary information.

JB: It's been my experience that a lot of *installed* enterprise software makes use of many of the peculiarities of their chosen database platform. For example, Oracle apps may use PL/SQL procedures, Oracle replication and/or SQLPlus automation commands. While PostgreSQL can provide equivalent functionality, the actual migration process is expensive and inherently risky.
This means that migrating is only feasible if either:
a) the application needed to be refactored anyway, or
b) there is an enormous amount of money to be saved.
For (b), in most of the "successful" migrations I've seen the company was saving more than $250,000 and 25% of their total software budget in license fees by switching to PostgreSQL.

LC: I've been wanting to play with firebird but just haven't had the time. I have heard very good things about it. I haven't heard of Fyracle. Interesting name. I'll add that to my to bottomless to do list.
If it has 95% syntax compatibility, what is its feature compatibility?

JB: Beats me. I just know about it because of the e-mail I get from the Firebird people.

LC: Here we disagree. XML DB, Streams, RAC, Grid, ADM, Bitmap Joins, Contexts, etc. The list goes on and on. That's not innovation? I think the only chance open source has to supplant Oracle, is Oracle's own stupid licensing.

JB: Well, Streams they got from OSS, same as anyone else (including us, when we get them integrated). As a hard-core relational geek, I regard XML-DB as a misfeature. RAC is kick-butt, but was developed 3 years ago (and is one of our main targets to supplant). I have yet to meet, or even hear of, anyone actually using Grid. Bitmap joins are nice and we'd like to have them for 8.2 or 8.3 -- they should be easy once we work out other varieties of index-only joins. I guess I should have said "Oracle isn't innovating very fast". Mostly it's because for every dollar they spend on R&D they have to spend $100 on market research, marketing, planning, legacy support, etc. What's ADM?
You're correct that the licensing is Oracle's doom. And there's not much they can do about it. As their market shrinks due to competition from OSS, IBM and Microsoft, they have to charge their existing customers more or fold. Which shrinks their market further. It's classic Clayton Christensen.

LC: I would hope OSS is innovative. That should be its strength. A lot of minds collaborating.

JB: There's a huge difference in speed. PostgreSQL puts out a new version every year, which includes major new features. The proprietary database companies do about one per 3 years or more because they have a planned development cycle. Hybrid companies, like MySQL, fall somewhere in between.

--Josh Berkus
PostgreSQL Project
(the opinions mentioned above are only my own)

LewisC 5/25/2005

Josh,

JB: Linux has been boosted by IBM spending $100 million marketing it. That wasn't Linus' doing that I know of. And we are working on promoting PostgreSQL; Google for us and you'll see a fair amount of press.

The problem with Googling for you is that if people don't know to look for you, they can't google for you. Does Linus do interviews? I think I've seen or read him in several venues. But I'll take the point that that's not the best example. How about Richard Stallman and GNU? Anyway, I'm not trying to argue. I just think it would be beneficial for Postgres to show up in the news mags more. Just my opinion.

As a hard-core relational geek, I regard XML-DB as a misfeature.

I have heard others make pretty much the same comment. Mind if I ask why? Is it XML in the database or the technology employed? XML DB is (or can be, I should say) purely relational under the covers but should I chose, I can access it in a variety of ways including SQL or XQuery. And for a system delivering content, as opposed to data, XML in the DB makes a lot of sense.

I guess I should have said "Oracle isn't innovating very fast". Mostly it's because for every dollar they *spend on R&D they have to spend $100 on market research, marketing, planning, legacy support, etc.*

I have to agree with that.

What's ADM?

Sorry, that was a typo. It should have been ADDM. Automated Database Diagnostic Monitor.

You're correct that the licensing is Oracle's doom. And there's not much they can do about it. As their market shrinks due to competition from OSS, IBM and Microsoft, they have to charge their existing customers more or fold. Which shrinks their market further. It's classic Clayton Christensen.

With all respect to Mr Christensen, I don't think Oracle's doom is imminent. Oracle needs to take some action now and be proactive keeping their customers happy (which means lowering prices) but they are innovating. I see nothing that makes me think they aren't still the most reliable database that exists. I'm not selling my stock yet.

JB: There's a huge difference in speed. PostgreSQL puts out a new version every year, which includes major new features. The proprietary database companies do about one per 3 years or more because they have a planned development cycle. Hybrid companies, like MySQL, fall somewhere in between.

Again I agree that Oracle has a longer developmenet cycle. I think Oracle gets more done per cycle though.

How long has PostgreSQL been in existence? How long has Oracle been in existence? Seems to me I've been hearing that PostgreSQL or MySQL was going to catch up to Oracle in 2 years since sometime in the early 90s.

My point is more that if Oracle doesn't start taking action, it doesn't matter if PostgreSQL is 75% or 90% or 100% of the functionality, companies will start abandoning them. I think in some cases, companies already are. But like you pointed out, only where it saves a tremendous amount of money and where the DB they're switching to is capable.

BTW, I'm glad you pointed out EnterpriseDB. I just found this article.

Time for me to start on my next article now.

Thanks again for the great discussion,

Lewis

Josh Berkus 5/25/2005

Lewis,

Anyway, I'm not trying to argue. I just think it would be beneficial for Postgres to show up in the news mags more. Just my opinion.

Hey, no question. You got any magazine publishers you want to introduce me to?

I have heard others make pretty much the same comment. Mind if I ask why? Is it XML in the database or the technology employed? XML DB is (or can be, I should say) purely relational under the covers but should I chose, I can access it in a variety of ways including SQL or XQuery. And for a system delivering content, as opposed to data, XML in the DB makes a lot of sense.

Well, it's a question of, is it XML stored in a relational database (which is keen) vs. XML used instead of relational apparatus (which is a waste of time). Most XML-DB stuff I've seen involves a hefty does of the latter -- a substitute for designing a proper database. If I hear the phrase "unstructured data" one more time I'll scream.

Heck, we have an XML data type although it needs improving.

Sorry, that was a typo. It should have been ADDM. Automated Database Diagnostic Monitor.

Looks nifty. Similar to something we're working on, but Oracle is further along.

How long has PostgreSQL been in existence? How long has Oracle been in existence? Seems to me I've been hearing that PostgreSQL or MySQL was going to catch up to Oracle in 2 years since sometime in the early 90s.

Not from me you haven't. I wouldn't say that our development momentum passed Oracle's until version 7.4.

Don't get me wrong, most of us have a lot of respect for Oracle in the PostgreSQL project. Oracle has a pedigree and history to match (or beat) ours and a real dedication to reliability and performance. If there's any RDBMS I enjoy competing with, it's Oracle. But, ultimately, I don't feel that any proprietary software company can match open source for speed of development, and I think comparing development track records over the last 3 years bears that out.

Thanks again for the great discussion

Thank you! I've quite enjoyed it.

Jutta Horstmann 8/10/2005

Interesting article and discussion, thanks.

I would like to add another link regarding comparison of Open Source databases:
http://www.osdbmigration.org
This leads to resources connected to my diploma thesis on "Migrating to Open Source Databases" as well as an Open Source databases feature matrix.

Another very good resource is Fabalabs Open Source databases comparison:
http://www.fabalabs.org/research/papers/FabalabsResearchPaper-OSDBMS-Eval.pdf

Regards,
Jutta Horstmann

Developing with Open Source and Oracle

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/005416.asp

Posted 8/18/2005

I was recently thinking about Oracle and PHP. Actually I was thinking about Oracle and open source in general. The link to the left there used to be to Oracle's Open Source Developer center. Anyway, that got me to thinking about Non-Oracle open source Oracle software.

I use a few like:

OraSchemaDoc: A JavaDoc like documenter for Oracle databases. Very handy if you use table and object comments. It also shows dependencies.

TOra: I am a TOAD user and TOra is the open source client most like TOAD. It's so close that TOra has been taken over by quest and incorporated into TOAD.

MOD_PLSQL: Has anyone doing any web development with Oracle not heard of MOD_PLSQL? It's an apache add-on for accessing PLSQL stored procedures from the web.

Computronix open source: Actually the only one of these that I have used is cx_Oracle, an interface to Oracle from python. I used it with OraSchemaDoc. This site is worth checking out.

DB Solo: A development tool with features for DBAs. This is shareware. I downloaded it and used it a bit to try it out. Looks good. I didn't hit any bugs. My company pays for TOAD though so I'll stick to that. A quality tool if you don't want to pay the premium for TOAD.

And here are a couple that I have downloaded and have the intention of checking out. I have the intention; having the time is a different story.

utPLSQL: This is a test framework for PL/SQL from guru Steven Feuerstein. I haven't seen any reviews of it but have a reproducible, template driven test harness would be beneficial. I really hope I get the time to check this one out.

Perl XSQL: An XSQL replacement. I used XSQL heavily a couple of years ago. I write several apps with it. I was disappointed in a few features and have been meaning to check and see if those issues had been fixed. I'll check Perl XSQL out also. Sometimes it's nice to have access to the source when you don't like a feature.

And here are a couple of Open Source tools that I use but aren't directly related to Oracle.
JSpell: Java Spellchecker for HTML. It's not free but it is a nice tool.

DIA: Completely free modeler. Can do UML and ERD. Awesome non-repository based modeling tool.

Gantt Project: Gantt Charts. No need to pay for MS Project. This is also a very nice tool.

That's all I've got. Do you use any open source tools or applications that might be useful to Oracle developers? Let me know.

Comments

chet justice 8/18/2005

Slightly similar to utPLSQL is SQLUnit. I used it at my previous employer and began using it home for various projects as well. It had REF CURSOR support which utPLSQL didn't have at the time.

Ravi 8/18/2005

Orabm and Orastress: TPCC like benchmarking open source tools [http://www.dbcool.com/]

//Ravi

LewisC 8/19/2005

Ravi and Chet,

Great. More tools that I won't find time to check out. heh Just kidding. I'll have to check those out.

I forgot two other open source tools that I use(d). I was using JEdit and just recently switched to Programmer's Notepad. There was nothing wrong with JEdit; I just like to switch tools on occassion.

My editor of choice is still Programmer's File Editor. Don't ask me why. It's an unsupported, freeware, closed source editor that hasn't been updated since 1999. For some reason, it's one of the first things I install on any new computer. It's just so fast and so easy to use I guess.

Thanks,

LewisC

James F. Koopmann 8/19/2005

Lewis,
I try and stick a completly free toolset. Why? I don't quite know other then I don't like to spend the money for large solutions when I can get things done better and faster the old fashion way.

Then again my editor of choice is vi. :)

Anyway I did put together almost a full month of reviews of Oracle related administrative tools in the February - March timeframe. For those interested just go to my blog home page at http://blogs.ittoolbox.com/database/solutions/ and then go to the Blog Archive area. Look in the March - February time frame. There are a lot of good free tools out there but my top 5 are here at http://blogs.ittoolbox.com/database/solutions/archives/003643.asp

Cheers,
James F. Koopmann

LewisC 8/21/2005

James,

So have you switched to primarily open source databases or did you just mean the tools you use?

Thanks,

LewisC

Chet Justice 8/23/2005

How could I forget JDeveloper? A tremendous tool which is now free. I use it for SQL/PLSQL editing (I like pretty colors) and some jsp development. Other than being a memory hog, it's a great tool. With the discovery of HTMLDB though, it will become solely an editor...

LewisC 8/24/2005

jDeveloper! I can't believe I forgot it too. Of course it's not open source, but it is free. I like it for source level debugging for pl/sql and java and for the xml editor. I haven't used the uml tools yet.

Thanks for the reminder,

LewisC

LewisC 8/24/2005

It's funny how you use something so often it disappears in your mind. Two open source tools I use on a daily basis are Filezilla for ftp and firefox for web browsing.

Everybody knows about firefox so I won't go into that but if you have a need for any ftp'ing, checkout filezilla. I used to use WarFTP but after finding filezilla on sourceforge, I've stuck with it.

Thanks,

LewisC

Peter 8/25/2005

Lewis,

Check this out - dbSaint at www.saintTechnologies.com. It is not open source but it costs only $149 and has most of the stuff in TOAD from Quest software. The company is also talking about making it opensource.

Cheers,
Peter

Alain Viret 8/31/2005

I'm using "Druid" (The Database Manager) which works not also for Oracle. With this tools, you may create a documentation of a schema in 5 minutes.

See http://druid.sourceforge.net/ and an extract from the sourceforge project page:

Druid is a GUI tool for database build and management. Users can add/change/delete DB objects (tables, fields, etc). Druid generates for you: SQL scripts, docs in XHTML, PDF, DocBook, etc; code in C, C++ & Java Beans even for JDO and support Castor & OJB.

LewisC 9/3/2005

Peter and Alain,

Thanks for the info. I'll check them out when I get a chance.

LewisC

Pranav Aggarwal 9/18/2005

Hi,
read ur blog, its good. hope fully u will be able to guide me.
I tried around 3 days to connect oracle with php but all in vein. lastly i had to work with mysql and working with it is also good, but mind is inquisitive that how oracle connect to php. i would appreciate if u could tell me .
thanks

LewisC 9/20/2005

Pranav,

I'm not a PHP guru so I'm not sure how much help I can give. Actually, that gives me a good idea. I think I'll learn PHP.

Anyway, did you check the FAQ on OTN?
http://www.oracle.com/technology/tech/php/htdocs/php_faq.html

There's a section on "How do I connect to Oracle".

Hope that helps,

LewisC

Ravi Ramakrishnan 10/2/2005

Another open source TPCC load generator

http://hammerora.sourceforge.net/

and they claim that this works against databases besides Oracle. This may be useful when you publish your "Oracle 10g vs PostgreSQL 8 vs MySQL 5 - PART 2".

Highway 11, Slidell, LA – December 23rd, 2005

Chapter 4 - EnterpriseDB

If I should die, think only this of me: That there's some corner of a foreign field that is forever England. - Rupert Brooke, The Soldier

EnterpriseDB - My First Look

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004863.asp

Posted 6/6/2005

Have you heard of EnterpriseDB? If not, check out these links:

Enterprise Tweaked PostgreSQL Solution - Oracle Killer?

EnterpriseDB Brings Open Source to the Enterprise Database

EnterpriseDB Fails to Ignite PostgreSQL Community

EnterpriseDB CEO: Open Source Databases are 'Unstoppable Force'

EnterpriseDB Targets MySQL, Unveils Its Own Open-Source Approach Live on SYS-CON.TV

EnterpriseDB to launch open source Oracle compatible RDBMS

The short story is that a company has taken the open source database PostgreSQL and made modifications to it to be more Oracle-like.

Well, after all of the buzz and word that this database is an Oracle killer, I had to download it and try it out. I also sent off for the free t-shirt. I would cheat and tell you what one of the demo table names is but I think you should download it and see for yourself.

An important note here: **This is a BETA release.** This is not a full production release. Expect issues and problems.

EnterpriseDB is touting it's Oracle compatibility. Their Oracle compatibility is their "redwood" mode, i.e. Redwood Shores CA, Oracle's headquarters. At this point there is no package support, no decode, and a lot more functionality is missing. FUNCTIONS are not supported in this release. I believe that will be fixed shortly. There are plans to support more Oracle bits but time will tell. Two things that I will say "Ahh. Oooh." about is when I see packages and autonomous transactions.

I did speak with Denis Lussier, Chief Architect and Chairman of EnterpriseDB Corporation, about when to expect packages. He said that we can expect a production release candidate 1 (RC1) in June and packages by Q3. He even mentioned global variables and global cursors. Nice! As a matter of a fact, by Q3 they also plan to allow explicit transaction control. Not quite autonomous transactions but they really seem to be looking to create a complete environment.

I think EnterpriseDB is interested in what the Oracle community thinks. Denis is an ex-Oracle DBA/Developer. He asked me, and I quote, "What else is EDB not thinking about that's a must have for fundamental Oracle compatibility that most (or at least many) Oracle applications use." Therefore, I pass the question on to you. What do you think is required to create a complete Oracle replacement? Post your thoughts and ideas here. I'll make sure they are passed on.

So on to the meat of this entry. I installed EnterpriseDB in a moment of open source madness. It installed flawlessly on my Windows XP machine. It asked very little. I accepted all of the defaults, which probably helped the install go smoothly.

EnterpriseDB comes with a utility, EDB-Studio. Kind of like Oracle's OEM but not quite as feature rich (or bloated depending POV). It features tree-oriented navigation. You can view all of the normal DB stuff: Schemas, tablespaces, users, roles, etc. Under the schema node, you can see tables, views, procs, triggers, etc.

One of the first things I noticed after looking through some of the sample procedures and triggers is that EnterpriseDB does support DBMS_OUTPUT.PUT_LINE. How very Oracleish!

I decided to see what kind of support they have so I browsed the user forums. As far as forums go, it's not very active. From a posting rate, I mean. It seems to be active by the employees of EnterpriseDB and questions are answered quickly. I posted a few just to test the waters and got answers within a day. But, alas, more about that below.

The documentation is so-so. There are enough new features that are different from standard PostgreSQL that there is a strong need for good documentation and that just isn't there yet. It is a beta though so I'll keep you posted on that detail. The basic docs that come with the install do have enough info to get you started if you are already familiar with databases in general and Oracle or PostgreSQL in particular. There is even a nice little tutorial schema to play with.

So, I decided to jump in and create a procedure. I created this one:

```
CREATE OR REPLACE PROCEDURE sample.parse_string(
                        p_instr IN VARCHAR2,
                        p_delim IN VARCHAR2,
                        p_len IN NUMBER )
IS

BEGIN

  DBMS_OUTPUT.PUT_LINE(
             SUBSTR(p_instr,
                    instr(p_instr, p_delim) + 1,
                    p_len)
             );

END;
```

This is pure PL/SQL (oracle's procedural language). I really didn't expect it to compile. It did. So I decided to write a wrapper for this procedure. I created this one:

```
CREATE OR REPLACE PROCEDURE sample.run_parse
IS
BEGIN
```

```
  parse_string( 'Lewis|Cunningham', '|', 6);

END;
```

I really didn't expect it to compile. It did. So I ran it from studio:

```
begin
  run_parse;
end;
```

It ran but it blew up. I got this fairly cryptic error:

ERROR: EDB-xxxxx: function substr(varchar2, integer, numeric) does not exist HINT: No function matches the given name and argument types. You may need to add explicit type casts. CONTEXT: SQL statement "SELECT substr($1 ,instr($2 , $3) + 1, $4)"

What the heck error code is EBD-xxxxx? Hope that gets fixed before the database comes out of beta.

After checking the docs, I found that I am passing in the correct parameters to both SUBSTR and INSTR. I have no clue what the problem is. I tried re-working the code thinking maybe it didn't like the instr embedded in the substr. No go on that.

So I posted a question on the discussion forum and opened a problem ticket. I did both right about 3pm on a Saturday, June 4. I got an automated response from the problem ticket within a few minutes. The automated response says I should get an update within 24 hours.

EnterpriseDB - Beta 2

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004863.asp

Posted 7/6/2005

I finally got a chance to take a brief look at Beta 2 of EnterpriseDB. The first thing I did was try the same procedures that I created in my last look, EnterpriseDB - My First Look.

Everything compiled and when I ran it, I got the same error as in beta 1.

ERROR: EDB-xxxxx: function substr(varchar2, integer, numeric) does not exist HINT: No function matches the given name and argument types. You may need to add explicit type casts. CONTEXT: SQL statement "SELECT substr($1 ,instr($2 , $3) + 1, $4)"

I expected to get the error but I thought it would have an actual error number now. I know they're working on that. I'm pretty sure that will be in place by v1.

I made the modification specified by the tech last time and the procedure ran. Here is the tech's explanation of the error:

Hello, Lewis

The problem you've run into is basically due to data type conversion of function parameters. Generally, the EDB string functions take and return data type INTEGER where any sort of count or position is concerned. In particular, the 3rd parameter of the SUBSTR function, the length, is of data type INTEGER, but the parameter, p_len in your function definition is NUMBER.

In section 11.2.4 "Procedure and Function Calls", of the EDB documentation, there is a table that shows the implicit data type conversions that EDB will do when calling a procedure. From this table, you will see that a NUMBER will not be implicitly converted to an integer, but must be explicitly cast.

In your example, the simple fix is to change the data type of p_len to data type INTEGER which I've done so below and re-executed your procedure.

The problem I have with this is that in PL/SQL, INTEGER is synonymous with NUMBER (Number(38) I believe). For someone converting from Oracle, there will be a lot of conflicts with this particular conversion. There are also a lot of other possible conflicts. The table of implicit conversions is now in section 9.1 of the documentation and it looks like it should have worked. I must be reading the table incorrectly. I'll have to query support about that and post an update later.

On the topic of support, I posted a couple of messages and got a response to both with in a day. I asked about autonomous transaction support and the support person said that they would be available in the fall. That is a nice feature that enables all sorts of other features like audit and error reporting.

The biggest thing in beta 1 that I jumped on was function support. Functions are now compilable and useable. I created this simple function:

```
CREATE OR REPLACE FUNCTION test_function RETURN NUMBER IS
BEGIN
  DBMS_OUTPUT.PUT_LINE('Hello World!');
  RETURN 1;
END;
```

And ran it with this:

```
DECLARE
  v1 NUMBER;
BEGIN
  v1 := test_function;
  DBMS_OUTPUT.PUT_LINE(v1);
END;
```

If you're familiar with Oracle and PL/SQL, you can see that that function is pure PL/SQL. I have a much stronger warm and fuzzy knowing that functions work.

The other biggie added with this release is replication support. I have not tried it yet. According to the press, it's based on the open source Slony-I project.

I have to admit, the more I use EDB-Studio, the more I like it. Very intuitive and speedy to get around. I've had no problems with it. EDB also comes with a more SQL*PLUSish interface called edb-psql.exe. You won't find this program from the menu. It's a command line tool. From the little I've played with it, I'd say it's also more primitive than sqlplus so I'm glad to have EDB-Studio.

The installation directories have changed. There are more directories and apparently a lot more functionality that in the previous beta.

The installation comes with tomcat5, which is new. It installs automatically, at least I don't remember agreeing to it. If I shut it down, EDB-Studio stops working. I'm debating whether or not I like that. In addition, a database like this may very well be installed on a machine that already has tomcat or another web server running. I need to think a bit more about this before I comment but I think you can see where I would be concerned.

Here is the techs reply when I asked about it on the support forum:

The EDB Admin Server provides ETL services through Webstart (utilizing Tomcat). Chapter 18 of the EDB documentation explains the EDB-ETL architecture and its usage in details. We also plan on adding a scheduler and other admin services to the Admin server in future releases. Also, the Admin server does not communicate externally. Please let us know if you notice otherwise and provide details. Thank you

And a blurb from the docs he pointed me to:

Have you ever wanted to transfer data from one JDBC source to another and do some kind of transformation during transfer, like: normalize non-normalized data; create artificial keys; execute SQL statements during, before or after transfer? Have you ever wanted to create a database in the following steps: create tables, load initial data, create indexes, create primary keys, and create foreign keys? If your answer is yes, you need this application: EDB-ETL!
18.1. Purpose

EDB-ETL loads data from a JDBC data source (database) into JDBC data target (database) and it perform many transformations defined in an XML file.

This application can process several different types of databases.

Possible JDBC source/target database types are: EnterpriseDB, MSSQL, MySql, Access, Excel, Csv, PostgreSQL, Qed, InstantDB, XML, BorlandJDataStore, Oracle, HSQL, McKoi, DB2, Sybase and Paradox database.

Also, EDB-ETL has several loading possibilities: creating databases, creating tables, inserting data into an empty database, inserting data in to a non empty database, updating constant columns, updating relation columns, updating columns with system time, updating columns with a user ID, executing every possible SQL statement, ...

Obviously that means they have big plans for this database beyond just a simple PL/SQL engine and it means I have a lot more reading to do. I will post in the future when I get more information about this feature. One thing I do worry about though is them biting off more than they can chew. I would be happier to have all of the PL/SQL functionality before any kind of application server, UI or ETL support. Although, if they can find a good ETL niche market, that would be good for their longevity.

And speaking of biting off more than you can chew, I read in the discussion forum that a SQL Server version will be available also. And in 8.0.2.10, there is already a Redmond catalog to go with the Redwood (Oracle) catalog. Redmond being Microsoft's HQ and Redwood being Oracle's HQ. Some of the objects on the catalog are sysindexes, systables, sysobjects. For those of you who use SQL Server, does that sound familiar?

When I installed Beta 2 (8.0.2.10), the install did not overlay or remove 8.0.2.9. They apparently were able to run side by side. After playing with both, I shutdown 8.0.2.9 and uninstalled it. My Beta 2 EDB-Studio stopped working. The uninstall must have deleted a DLL it needed. The error suggested that I try to reinstall but the installer said this version is already installed. This is the point where I remind you, and myself, that this is a beta and is exactly why you don't put your production data in a beta database.

I opened a ticket with support to see what they would say. My guess was a reinstall and support confirmed that. Uninstall and reinstall. They did say this error would be fixed in the next release. And they did reply within 24 hours. I stress the 24 hopurs because that's pretty amazing for a beta product.

My opinion at this point is that EnterpriseDB is a work in progress and it's definately heading in the right direction. I've exchanged many emails with Andy Astor, the CEO, and Denis Lussier, the Chief Architect and even a couple of other techs and in all cases I find them open to feedback and looking for ways to improve.

And that's it for this overview. I'm under a serious time crunch at the moment but as soon as I get some free time, I plan to really put the screws to this database and see what it can do. Since it's based on PostgreSQL, I already know it will be robust. I plan to implement my own mini-version of AQ just to see if I can. I haven't heard about about the availability of DB Links though. Look for an entry about that in the near future.

EnterpriseDB General Availability

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/003611.asp

Posted 8/12/2005

EDB 2005 is now in general availability. I downlaoded it Wednesday night and installed it Thursday.

Doesn't seem like much has changed. Package support is still not there but it should be making it in by Q4. This looks like it's mostly a bug fix release.

What I do like is that they've released their pricing.

EDB Basic is free. Basic is available for development and low volume production. The definition of low volume production as 1 CPU, 4GB of data and 1GB of memory. I can understand the 1 CPU and 1GB of memory but why the 4GB of data? That's like crippleware that won't let you print or save changes.

Anyway, basic includes read only access to the web based support forums. That's pretty much no support at all. I expected that basic users would at least be able to post questions in the forum. The forums seem to me to be a fairly easy way to build the community as well as offer realtively cheap support. I'm disappointed in this decision.

EBD Silver is $1,000 per CPU per year. Silver support includes unlimited email and web based forum support with a 24-hour turn around (business days). Also, phone support from 9am to 8pm EST.

EDB Gold is $3,000 per CPU per year. Gold includes silver support with 4-hour web based/email turn around (business hours only) and 24/7 phone support. So, you get better support via phone on the weekends. Interesting.

EDB Platinum is $5,000 per CPU per year and includes everything Gold has plus email support in 1-hour (still business hours only). You also get a dedicated account manager, a perpetual license and IP indemnification.

My first question is about business hours. Does that mean 24-hours on a business day or 8am to 9pm EST? Or something else?

My second question is about the perpetual license. Que? Does that mean I can use it forever as long as I drop $5,000 per year per CPU? I thought a perpetual license gave you access to the software forever for a lump sum?

Third, I really would like to know where that 4GB of data in the basic license comes from. In the real world of 2005, 4GB is a starter database.

I left a message on the, still free, web forum asking these questions. I'll post an update when I hear back.

Update: 08/12/2005 1300: From EDB CEO Andy Astor -
A perpetual license means that you have perpetual rights to use the software, even if you stop paying for support.

Email support is always 7x24. Telephone support is business-hours only (9am-8pm) for Silver subscribers.

Comment from LewisC. *That makes sense on the perpetual license. That option didn't occur to me.*

So overall, what do you think of the pricing?

Here's my 2 cents. This is a product based on an open source database. Except for basic support, I think the dollar amounts are very fair. This isn't the model of support for a fee. EDB is an enhanced version of PostgreSQL supporting some very complicated features.

When they put out a version with package support, it should be fairly easy to recompile Oracle applications. That will mean a viable alternative to Oracle with minimal changes for a reasonable price.

I still don't see EDB as a replacement for VLDB or super high rate OLTP but there are plenty of uses where an Oracle compatible DB would be nice but the application just isn't worth an Oracle-sized license fee.

What I am disappointed in is the basic support. Come on guys. This is based on an open source project. Keep the web-based forum open on a 24-hour turn around at least. I can understand no phone support. People need to get paid. But, having no support is not really an option for most businesses; not even for low volume production applications. The web based forums are a great way to provide support that won't break the bank.

I even found this text on the web site:

EnterpriseDB 2005 is the answer for companies that need an enterprise-class relational database with enterprise-class support, but without the enterprise-class price tag.

Other than that, I'm still waiting for my packages. If any of you have also evaluated EDB, post a comment and let me know what you think.

Comments

Ravi Ramakrishnan 8/19/2005

Sorry, LewisC. But I have to rant.

"EnterpriseDB 2005 is the answer for companies that need an enterprise-class relational database with enterprise-class support, but without the enterprise-class price tag" sounds like a bad slogan. It is not going to get them anywhere.

Like you, I like Oracle. I do Oracle for a living. But, I want to see open-source databases succeed. Just like Oracle is giving the hardware vendors nightmares by pushing Linux deeper into the enterprise, we need a movement to present better alternatives to Oracle. Its too pricey and comes with a lot of legacy baggage. The path to the enterprise is not thru slogans but thru proof points. While price and features are important metrics, you dont lead with them. The reality is that the decision-makers consider open source as an alternative mainly because of price, but that will not be the reason they entertain them into the enterprise. We dont need to talk about VLDB at this point, instead show me a VSDB that was migrated seamlessly and showed substantial benefits.

You are probably asking me to stop complaining and start suggesting something that may open up a debate. I suggest:

1. We publish a TPC benchmark. Does not have to be the best in terms of TPMs but will easily make the top of the chart in terms of $/TPM. As a part of this process, we will learn a lot. It will help us strategize our development efforts. Most importantly, it will enjoy a lot of visibility among decision makers. Just imagine how much visibility Oracle gets from benchmarks. Anyone who dismisses this suggestion with "benchmarks are not practical" just dont get it. That is the starting point to any discussion on database and hardware selection. Besides, if one cant run the TPC scripts against your DB.........no one needs that DB.

2. Stop targeting big databases and complex applications. Terrible strategy. I had been a part of hundreds of discussions with customers exploring different hardwares, OS'es, databases and applications. Almost every customer has non-mission critical databases, small databases, non-production environments. If their expertise is in Oracle, they tend to run these other environments on Oracle, too. What a waste of time and money. The path to the datacenter is by targeting these other environments. Customers develop confidence and expertise by migrating these lesser critical databases before they consider the big one.

3. Development tools are great, but migration tools are more important. Enough said about this.

Next time I see someone presenting a slide comparing Oracle downloads with those of MySQL or EDB, I will throw up on that loser.

//Ravi

LewisC 8/21/2005

Ravi,

Next time I see someone presenting a slide comparing Oracle downloads with those of MySQL or

EDB, I will throw up on that loser.

Don't be shy. Tell us what you really think. ;-)

Seriously though, you raise some very good points. The TPC benchmarks are a great idea. I wonder if they are working on that?

I also agree 100% on your second point. It will be a long time before EDB has the credibility to take on the large and/or complex databases of entrenched Oracle customers. If they can get in and takeover the lowend, they'll have the foot in the door for the future. I think the biggies at EDB already know that though. They've got some heavy hitters on their managment team and I know they've put a lot of thought into where they want to go and who they want to be.

And on point 3, if they can fully implement the pl/sql language in EDB, they are a good way towards that migration tool, i.e. the language itself.

Jester, French Quarter, New Orleans, LA – Christmas Day 2005

Chapter 5 - Distributed Oracle

I do not fear computers. I fear the lack of them. - Isaac Asimov

Oracle Advanced Replication: A Definition in Plain English

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/003611.asp

Posted 3/24/2005

I'm working on the Part 2 to my Day with Ralph Kimball entry. I forgot my notes at work last night though so I'm posting this entry today instead. I'll get a chance to work on part 2 this weekend.

Today's entry is about Oracle Advanced Replication. This is an explanation and description of replication, not an implementation guide or a best practices document.

What is Replication?

According to Webster's Online, the definition we mean is: 3 a : COPY, REPRODUCTION b : the action or process of reproducing

Replication is simply copying data from one database to another. This could be a subset of the data or a complete copy.

I'm going to use insert for my examples but any DML can be replicated.

If you create an insert trigger on a table and push data across a database link every time a record gets inserted, you've performed replication. Call that the poor man's SYNCHRONOUS replication. If in that trigger, you populate a local queue for a later push or pull to a different database, that's ASYNCHRONOUS replication.

Oracle offers both types. The down side to synchronous replication is that the act of inserting a record is delayed until the data can be moved to all replicated databases. If a replicated database is down, the transaction cannot complete. The pro for synchronous is guaranteeing the transaction's delivery to all databases and the elimination of conflicts (more about that later). The downside to asynchronous replication is that depending on database availability, network bandwidth and other factors, the delivery of the transaction may be delayed. This delay could impact reporting or other functional areas. It also increases the likelihood of conflicts in an active/active environment.

A master database is a database actively taking DML. Active/active means that multiple databases are running and accepting transactions and are replicating that data to each other. Oracle calls this multi-master replication. I'll talk about that below.

So, what is Oracle Advanced Replication?

Oracle offers two types of replication: snapshot and multi-master. I'll tackle snapshots first.

Oracle offers a feature called snapshots that is synonymous with materialized views. A snapshot at it's most basic is a query that has it's data materialized, or populated. When you create a snapshot, a table corresponding to the query's column list is created (as is some other metadata not discussed here). When the snapshot is refreshed, that underlying table is populated with the results of the query.

For replication, as data changes to a table in the master database, the snapshot refreshes as scheduled and moves that data to the replicated database. Snapshot replication is for read-only replication. There are some instances of updateable snapshots but I have never seen that feature used in the real world. If you have an example of real world use, post a reply and describe it. I'd like to hear about it.

For multi-master replication, an application can be installed on two (or more) oracle instances, the common example being one on the east coast and another on the west coast. Both may have people entering data and that data needs to be kept in synch. Hence, they are both master sites and the phrase multi-master. This obviously is more complex than snapshot replication. There are considerations to keep in mind when using multi-master replication. The three most important, from my point of view at least, are sequences (which cannot be replicated), triggers (which can turn recursive if you're not careful) and conflict resolution.

Some people use multi-master replication as a failover device. That is NOT what replication is for and I would argue against it. Oracle provides other facilities specifically for failover. You can implement replication as a cheap psuedo failover but remember that you get what you pay for. By using replication in this manner, you will increase maintenance over time, both in conflict resolution and in system downtime.

How does it work in 8i, 9i, 10g? If definitely remember that replication was available in Oracle 7.3.4 but I'm not sure when it started exactly. In 7.3, replication was very basic, very limited and very manual. As far as I remember, synchronous replication has worked pretty much the same in all the versions: Triggers on the tables doing a push using db links. In later versions the triggers were hidden behind the scenes. Due to it's limited use, the discussion from this point forward deals only with asynchronous replication.

I'm not sure of the exact facilities in versions prior to 8 but in 8.0, Oracle created triggers and packages and dropped replicated DML on a queue table (using Advanced Queuing or AQ. I'll talk about AQ in the future). In 8i, Oracle hid the triggers and the queues. Later versions use streams and re-do logs. Streams is a topic for the future. The re-do logs keep a copy of everything done in a database. By reading the logs instead of using triggers, the overhead to the transaction, and to the database, is minimized.

And finally, what is conflict resolution? There are 2 kinds of conflicts that will usually arise in any replicated environment, key conflicts and update conflicts.

A key conflict is where a primary key is inserted into one instance and the exact key is inserted into another instance before the first can be replicated. Now which one do you keep? Or do you keep both but change the key? If you use sequences, you can run into this because sequences are not replicated by oracle (mainly for performance reasons, post a comment if you would like a more detailed explanation of why this is so). A way to avoid this is to use sequences, but prepend a DB identifier to the sequence, i.e. database A would have an ID of A and database B would have an ID of B. When a sequence is generated in A, the A is prepended to that 00000001 would become A00000001. Any that come in on B would get a B prepended. That would remove key conflicts. You could as easily use a number instead of a character, keeping the column purely numeric.

An update conflict is more difficult to fix. What happens when a record is updated on one instance but deleted on another? Or it's updated on both but in different ways? Who wins? Update conflicts are very application dependent.

Here's a scenario from a real production environment. An application was set up with 4 way (4 instances) multi-master replication. It was decided to use replication in this way for fail over. One of the databases took a real-time update at exactly the same time that a batch process on a different instance updated about 50,000 transactions. The real-time update made it to the other 2 instances and committed. Because the original transactions in the other 3 databases were changes, the 50,000 transactions failed on replication. Unfortunately, the batch process was gospel so now the other 3 databases had to be brought back into synch. This application had to come down for 12 hours while the data was manually re-synched.

The moral here is not to stay away from replication. I'm saying don't use it for fail over and however you do use it, plan for conflicts. Advanced Replication does provide some facilities for conflict resolution. The provided functionality wouldn't have helped the above scenario but if you have the situation where the last transaction in wins, or the first, or a particular database is gospel, you can use those to automatically resolve the update conflict. IF you plan to use replication and design it into your application from the beginning, you can be much more successful with replication. With a cut and dried set of conflict rules or by implementing replication after the fact, you have to plan on downtime.

In a warehouse environment, you can use replication, snapshots are nice for this, to move data from your OLTP database to you staging and/or warehouse database. You're ensured that all data will make it because Oracle guarantees delivery. This would be the E in ETL with no coding at all. You wouldn't ever have to worry about conflicts since you aren't updating the warehouse copy of the tables. From there it's very easy to fire up Oracle Warehouse Builder (OWB) to finish off with the T&L.

Well, I guess this is my first real beginning on my Oracle Encyclopedia. I've used some words and phrases you may not be completely familiar with: insert, transactions, sequence, fail over, etc. How would you like me to handle these kinds of definitions? Should I include a glossary in the article? Should I provide a definition in parenthesis whenever I use a new word someone might not understand? Should I immediately follow an article with a definitions document? Any other ideas? Post a comment and let me know your thoughts and as always, if you agree, disagree, or just have some thoughts to share, please post a comment.

Advanced Queues and Streams: A Definition in Plain English

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/003677.asp

Posted 3/30/2005

A mathematician, an accountant and an economist apply for the same job. The interviewer calls in the mathematician and asks "What do two plus two equal?"

The mathematician replies "Four."

The interviewer asks "Four, exactly?" The mathematician looks at the interviewer incredulously and says "Yes, four, exactly."

Then the interviewer calls in the accountant and asks the same question "What do two plus two equal?" The accountant says "On average, four - give or take ten percent, but on average, four."

Then the interviewer calls in the economist and poses the same question "What do two plus two equal?"

The economist gets up, locks the door, closes the shade, sits down next to the interviewer and says

"What do you want it to equal?"

And now for something completely different:

What is AQ? Advanced Queues, or AQ, is Oracle's messaging solution. AQ provides a persistent (or non-persistent for specialty apps) queue mechanism that can guarantee delivery of a message. It has interfaces to PL/SQL, OCI and Java. It's Oracle's answer to IBM's MQ Series.

A message can be an XML document, a set of fields, an array of data and just about anything else you can think of.

AQ works on a publish/subscribe model. That means that someone (a publisher puts a message on the queue and someone else (a subscriber) takes the message off. A queue can have multiple subscribers. Technically it can have multiple publishers but I haven't worked with that configuration and I'm not sure what the usefulness of that is. I think I would prefer multiple queues, one for each publisher.

What's AQ good for? What does it do?

One example would be replication. In Oracle Advanced Replication, AQ is the mechanism that copies data from one instance to another. The master site (publisher) will receive an update, it puts the update and a before and after image of the data in a queue. The slave sites (Subscribers) pull the data off the queue and apply them to the local database. The before and after images are used by replication to find the correct record and see if there are any update conflicts.

Besides Oracle replication, or your own home-grown replication, there are a lot of other uses for AQ.

You can drop a message on a queue for local usage. Say if you have a transactional system and are getting backed up but don't want to turn away incoming transactions. You can implement a queue. The receiving procedure can drop the transactions on the queue and a local de-queue procedure in the background can pull them off when it has time.

You can use AQ to interface with Java. AQ supports the Java Messaging Specification (JMS) API. Using Java, XML and AQ you can easily implement a SOA (service oriented architecture) web service.

What is Streams? Here's a brief description of streams and what you can use it for.

AQ and Replication both entail data movement. Streams is the current technology enabling that data movement. Streams is kind of like AQ, but with rules applied.

Let's think about AQ. AQ is basically a table and some table maintenance code wrapped around streams. When you enqueue a record, you're using AQ. Streams takes over and moves it to the next database, enqueueing it locally. AQ then takes over again, dequeuing it for consumption.

Streams has some nice features. I think the most important is the rule based transformations. A transformation allows you to modify the payload in flight. A receiving application doesn't need to be aware of the sending applications formats, it just receives what it needs.

Think of the way mainframes send data down to a data warehouse. The mainframe doesn't send entire vsam files down to let the warehouse figure out what pieces it needs. The warehouse group defines the fields it needs, a mainframe programmer writes a Cobol (maybe) program and sends a new, specific file down.

With streams, the receiving application can define what it needs and the sending application can define rules to match. The nice thing with streams is that there can be multiple consumers receiving the same payload but have different rules applied for them. The sender sends one payload and it's transformed many times in different ways for multiple consumers. Rules are also easily defined as opposed to writing a program, scheduling a batch processing window, writing a load routine, etc. With streams, identify the source, define the rules and write a consumer dequeue.

Speaking of data warehouses, another use of streams is in change data capture (CDC). You identify a source object, say your transaction detail table in the oltp system. You can create a rule that says capture all transactions that are approved and billable. Define the billing warehouse as a consumer for that stream. That payload can be applied to a staging table for loading into a warehouse table. The apply to the staging table can be done without coding. With almost 0 lines of code you can move the data you want, i.e. billable items, from your oltp system directly to your warehouse.

Streams is also non-stressful to the source database. Streams reads the redo-logs and gathers information from that as opposed to running queries or DML against the source database's tables.

Streams and AQ are both pretty fascinating technologies. And there is a lot more to them than I speak about here. In the near future I want to show how to do some setup and build a little application using CDC. If you're using AQ in 10g, then you're already using streams behind the scenes. I think that's the hallmark of good technology; it makes life easier and you never have to see it.

Database Links: A Definition in Plain English

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/007023.asp

Posted 12/21/2005

What is a Link?

A database link is the way Oracle allows you to connect from one database to another. There are several types of links and I will discuss each of those. Database links are also used in heterogeneous access and I will, very briefly, describe that.

Database links enable distributed processing. Replication is an excellent example of an application using database links. Oracle replication uses database links to move data from one database to another. For a description of replication, look here.

A database link is a one-way conduit allowing you to perform DML against remote databases just like you would against a local database. Your SQL does not need to change except to qualify which database you wish to access.

A discussion of database links can quickly degenerate into a discussion of both SQL*Net and security. I am purposely staying away from that so that the discussion remains about database links. Now, having said that, I am going to say two things:

First, any discussion of implementing distributed access in a production environment should include discussions about security. Do it wrong and you have a bad thing. Oracle provides several methods and you need to choose the one most appropriate for your goal.

Second, database links require the remote database to be defined to your server and not in your client configuration. Your client may know nothing about a database and you will still be able to create a link. The reverse also holds true; if your client is aware of a database but the server is not, you cannot create a link.

Type of Links

- **Connected User** - This link always connects as the currently connected user. This requires the user using the link to have an ID in both databases. When using a connected user link, your access is in the remote database is defined by the privileges of that ID in the remote database.

  ```
  Syntax: CREATE DATABASE LINK [linkname] USING '[remote_database]';
  ```

- **Current User** - Current User uses a global user; that is, a user authenticated externally via a directory service.

  If using a current user link within a non-stored object (i.e. from SQL*Plus, Pro*C or JDBC), a current user link will use the login credentials of the logged in user. So if BOB create a current user link and TOM uses that link, TOM's credentials in the remote database will be used. That means that BOB and TOM must both be globally defined in, and have access to, the local and remote databases.

  If using a current user link in a stored object (i.e. a view or stored procedure), the link will use the login credentials of the user who compiled the object unless the object uses invoker rights.

  This method is very secure and is part of Oracle Advanced Security. This allows a less privileged user to run a remote procedure with a more privileged users access.

  ```
  Syntax: CREATE DATABASE LINK [linkname] CONNECT TO CURRENT_USER USING
  '[remote_database]';
  ```

- **Fixed User** - Fixed user is probably the most common type of link. A fixed user link connects via specific credentials. Every time the link connects it connects with the same user ID and password. The fixed user ID must be defined in the remote database but does not need to exist in the local database.

  ```
  Syntax: CREATE DATABASE LINK [linkname] CONNECT TO [username] IDENTIFIED BY
  [password] USING '[remote_database]';
  ```

- **Public** - A public link is one of the above links created with the PUBLIC keyword. PUBLIC links are viewable by anyone. Non-public links are private links and can only be accessed by the schema that created it.

- **Shared** - A shared link is not really a different kind of link. It defines the way Oracle manages the physical connections to the remote database. A shared connection can reuse connections. Shared is useful in appropriate circumstances but is not appropriate for all circumstances.

- **Global** - A global link creates a network-wide link. If you are using a directory server and your databases are identified by net names, you can create global database links. Every database identified by your server will get the link. This makes managing the links easier for administrators.

Heterogeneous Services

Heterogeneous services give Oracle the ability to access non-oracle data sources. Oracle uses database links to present the foreign database to the user. Oracle also presents a data dictionary translation to the user.

There are two kinds of services: Transparent Gateways and Generic Connectivity.

Transparent Gateway

A transparent gateway is connectivity to a non-oracle data source that has been optimized for the database. That is, the IBM DB2 transparent gateway will use native DB2 connectivity to access the database. If there were a gateway, this would be the best way to approach accessing that data.

Generic Connectivity

Generic Connectivity is a generic transparent gateway; only it's not so transparent. Heh If you need to access a database that does not have a gateway and is accessible via ODBC or OLEDB, generic connectivity is the method to do so.

Plain Text is not considered a non-oracle data source. If you want to access text as data, use external tables.

Examples Using DB Links

Select sysdate from the dual table in a remote database names RemoteDB using a connected user link:

```
CREATE DATABASE LINK MyRemoteDB USING 'RemoteDB';
SELECT sysdate FROM DUAL@MyRemoteDB;
```

Execute a stored procedure in the remote database:

```
BEGIN
  runProc@MyRemoteDB;
END;
```

Execute a packaged function in the remote database:

```
DECLARE
  var1 VARCHAR2(10);
BEGIN
  Var1 := runFunc@MyRemoteDB;
END;
```

Create a view using the select above:

```
CREATE OR REPLACE VIEW Remote_Sysdate_VW AS
  SELECT sysdate AS remoteSysdate FROM DUAL@MyRemoteDB;

SELECT * FROM remote_sysdate_vw;
```

Conclusion

If you need to access procedures or data in two or more Oracle databases, database links are the way to go. If you need to access non-oracle data with SQL or PL/SQL, use a database link via heterogeneous services.

There are a lot more considerations for database links than those I bring up here such as naming, domains, global names, security, maintenance, dependencies, two phase commit (TPC), etc. In the future, I would like to address each of these. If any of these are more important to you or you would like to see discussions on these first, please post a note and let me know.

Accessing Multiple XE Databases from a single machine

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006839.asp

Posted 12/8/2005

I was asked by a friend of mine how he could access the XE databases that he has. There's one on a server, one on a co-worker PC and one on his desktop. All of them are called XE. He's Oracle literate but not a DBA or developer and was not comfortable messing with much of his configuration.

My suggestion was to just modify his TNSNAMES.ORA file so that his net service names were different. He had three entries like this (he cut and pasted from each of the machines):

```
XE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = myDesktop)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )

XE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = otherDesktop)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )

XE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = sharedServer)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )
```

So, I had him change it to look like this:

```
myXE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = myDesktop)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )

thatXE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = otherDesktop)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )

serverXE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = sharedServer)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )
```

I also walked him through creating a couple of database links to test and make sure everything worked.

```
CREATE DATABASE LINK thatXE
  CONNECT TO HR
  IDENTIFIED BY HR
  USING 'thatXE'
/
```

and

```
CREATE DATABASE LINK thatXE
  CONNECT TO HR
  IDENTIFIED BY HR
  USING 'serverXE'
/
```

Those worked and he was up and running. That was about the simplest thing I could think of. Can you think of anything as easy or easier to accomplish the same thing?

BTW, I asked him what the databases were for. The two desktop copies are for them to learn HTML DB and the server database is eventually going to be a test server. Production will probably an SE server that they already have running some other apps.

I just think it's so cool that an intelligent person, but not a DBA or developer, was able to get the Oracle XE software, install it and have it running without any issues. The XE and HTML DB folks at Oracle have done a terrific job on this.

Since I had to explain about database links to my friend (that was actually the most complicated part of this to him), I decided to write a paper on database links. I'll post that over the weekend or possibly, on monday. (The DB Links areticle was written in 2006 so it's not included here.

City Park Tent City, New Orleans, LA – December 24th, 2005

Chapter 6 - XML

XML is like violence - if it doesn't solve the problem, you aren't using enough of it.

XML in the Database: A Brief Overview

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/005319.asp

Posted 6/2/2005

Oracle provides a lot of XML functionality in the database. This entry will cover the different tools Oracle provides and future entries will get into the details of each.

The three primary XML tools that oracle provides are: XML Developers Kit (XDK), XML DB and SQL Extensions for XML (SQLX and XMLType). Below is a definition of each.

XDK

The XDK is a complete XML development toolkit that was introduced for 8i. I think the first version was labeled XDK9i but the toolkit actually was available for 8i. I remember using the beta product. ;-)

XDK has everything you need for developing complete XML applications in the database. It supports DOM (tree access) and SAX (streaming access); and JAXP if you prefer implementation independence. It also has an XSLT processor.

The XDK also comes with the XML SQL Utility. XSU is the API allowing creation and storage of XML. DBMS_XMLQuery.getXML accepts a query as a parameter and returns an XML document.

DBMS_XMLSave.InsertXML allows a program to insert XML documents into tables and columns. The XML must conform to some limitations, i.e. element tags must match column names. That is somewhat configurable. You can also update data with XML documents by using DBMS_XMLSave.UpdateXML and delete by using DBMS_XMLSave.DeleteXML.

Personally, I believe that there are better ways to extract XML from the database than DBMS_XMLQuery and better ways to get it in than DBMS_XMLSave. If you're using Java or C and you are presenting XML on the web, the XDK is the way to go but if you're using PL/SQL, I really think Oracle made it more complex than it needs to be.

The XDK also provides the XSQL servlet which is a nice, web based XML access and presentation utility and they provide a SOAP framework. Both of these fall under the XDK header. XSQL is basically a utility where you pass in a SQL query and an XSLT document and it will present it on the web. It is somewhat programmable but I found it too limited for real application use. For a simple in house reporting solution or something along those lines, it is functional enough.

XML DB

XML DB is an XML repository for XML. This repository gives high performance access to XML documents and let's you store those documents in a seemingly unstructured format (but is really relational under the covers). XML DB ties in well with the XDK to create Java applications that create and store XML documents.

With XML DB you have a folder (directories, sub directories, etc) view of your documents. You can grant permissions to documents as you would on a file system and can even keep versions of documents.

Oracle has added FTP and HTML access to the database to support XML DB. You can configure listeners to listen on those ports and have the database act as a server.

XML DB is a great tool when you need to save, modify and present XML content. The modify word is the important part. If all you are doing is generating and presenting XML, there is no reason to store it as XML. If the data is already XML and you are just presenting it, there is no reason to put it in a database. If you have XML and need to add to it, update it or delete some of it before presenting it, or maybe even merge documents selectively (think where clause), XML DB is where you want to do it. What makes XML DB nice it the choice to use it as XML or to access and modify it via SQL.

The power of XML DB, in my opinion, is the support of advanced SQL XML queries and the new XMLType. I use these tools on a day-to-day basis.

SQL Extensions for XML

The XMLType is an OO XML aware data type. It can be used in columns or in PL/SQL just like VARCHAR2 or DATE. XMLType has member functions that allow access to data using XPath.

A quick example extracting a specific value from an XML varchar2 string:

```
DECLARE
  v VARCHAR2(32000) :=  '<DATA>
                             <LINE>A</LINE>
                             <LINE>B</LINE>
                             <LINE>C</LINE>
                        </DATA>';
  x XMLType;
BEGIN

  x := XMLType(v);

  DBMS_OUTPUT.put_line(
           x.extract('/DATA/LINE[1]').getStringVal()
                 );

END;
```

If you read my previous entries on OO, you can see that XMLType is an Object type. The first line after the begin is the XMLType constructor. In the DBMS_OUTPUT, I'm extracting the first child node and displaying it as a string. XMLType has many useful built-in functions. I only dreamed of such usefulness in 8i. At times, in 8i, manipulating XML was down right painful.

SQLX is a set of SQL functions that give a programmer the ability to generate XML documents and fragments. There are a handful of new XML* functions described in the SQL reference. The ones I'm finding to be most useful are XMLElement, XMLAGG and XMLForest.

Here is a quick sample using SQLX:

```
SQL> SELECT
  2      XMLElement("Ename",ename ) Ename,
  3      XMLElement("Empno",empno ) Empno
  4    FROM EMP
  5    WHERE rownum < 5
  6  /

ENAME
------------------------------------------
EMPNO
------------------------------------------
<Ename>SMITH</Ename>
<Empno>7369</Empno>

<Ename>ALLEN</Ename>
<Empno>7499</Empno>

<Ename>WARD</Ename>
<Empno>7521</Empno>

<Ename>JONES</Ename>
<Empno>7566</Empno>

SQL> SELECT
  2      XMLElement("Employees",
  3     XMLForest( ename, empno ) )
  4    FROM EMP
  5    WHERE rownum < 5
  6
SQL> /

XMLELEMENT("EMPLOYEES",XMLFOREST(ENAME,EMPNO))
---------------------------------------------------------------
<Employees><ENAME>SMITH</ENAME><EMPNO>7369</EMPNO></Employees>
<Employees><ENAME>ALLEN</ENAME><EMPNO>7499</EMPNO></Employees>
<Employees><ENAME>WARD</ENAME><EMPNO>7521</EMPNO></Employees>
<Employees><ENAME>JONES</ENAME><EMPNO>7566</EMPNO></Employees>
```

And that's just a small part of the power of SQLX. The Oracle 10g SQL reference and XML DB manuals are great resources on this. This overview is the segue to my next article, The Sorcerer of OO Part 4, where I will combine XML and Object Views. In the coming weeks I will also write in more detail how to, and when to, use XML in the database. I will cover SQLX and XMLType in painful detail.

Oracle and XML In Action - A Real World Example

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/005319.asp

Posted 8/10/2005

I've written some in the past about Oracle and XML. A couple of days ago, some coworkers had a requirement and I created a simple example to show them one possible solution. I thought I would share that here.

The requirement was to have a stored procedure accept an XML document and insert it into a table. The XML document was fairly complex and contained a lot of information not required by the application.

A fairly simple requirement but one that could be written in many ways. We could have done a java stored procedure or called the PL/SQL interface to the DOM processor or any one of many other possibilities.

I like to keep things as code free as possible. The less code I write, the less code I have to maintain. Oracle provides an abundance of built-ins and the hardest task we have when designing an application is deciding which approach to take. This is my approach to this particular requirement.

I am keeping the XML and the table small. I want to make what's going on as clear as possible. The XML document my coworkers were using had, I think, a couple hundred elements and they wanted about 50. I'm using an XML document with 4 elements and I'll only keep 2.

First, let's create a sample table:

```
CREATE TABLE tran_data (
  tran_id NUMBER,
  rcode VARCHAR2(10) )
/
```

This is the table we want to populate.

Here is the document we have to start with:

```
<TRAN>
  <TRAN_ID>1</TRAN_ID>
  <RCODE>A</RCODE>
  <TRAN_DT>01-JUN-2005</TRAN_DT>
  <STATION>1234567</STATION>
</TRAN>
```

So how do we get that XML into that table? Like I said there are many ways to do it. I will actually show two ways below.

The first is to create a generic procedure to move an XML document into a table by converting the elements to Oracle Canonical format.

Oracle's Canonical format is:

```
<ROWSET>
  <ROW>
     <column_name_1 />
     <column_name_2 />
     <column_name_3 />
  </ROW>
  <ROW>
     <column_name_1 />
```

```
      <column_name_2 />
      <column_name_3 />
  </ROW>
  .
  .
  .
</ROWSET>
```

ROWSET identifies the overall XML document; the ROW tag is repeated for each logical row. Elements within the ROW tag must correspond to actual table column names. ROWSET is an optional tag if you only have a single row. It is required for multi-row processing.

So, how do I get from my example above to a canonical format? Let's use XSL. I can transform the above XML using this:

```
<?xml version="1.0"?>
<xsl:stylesheet
               xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
               version="1.0">
  <xsl:template match="/">
    <ROWSET>
      <ROW>
        <TRAN_ID>
          <xsl:value-of select="TRAN/TRAN_ID" />
        </TRAN_ID>
        <RCODE>
          <xsl:value-of select="TRAN/RCODE" />
        </RCODE>
      </ROW>
    </ROWSET>
  </xsl:template>
</xsl:stylesheet>
```

So, now the idea is to use the XSL to transform a standard XML document in a canonical format.

What da ya know? Here's a procedure to do just that:

```
CREATE OR REPLACE PROCEDURE xml2tab(
                  p_xml_in XMLType,
                  p_xsl_in XMLType,
                  p_table IN VARCHAR2 ) AS

  v_context DBMS_XMLStore.ctxType;
  v_rows NUMBER;

BEGIN

  --Open a new context, required for these procedures
  v_context := DBMS_XMLStore.newContext(p_table);

  -- This is the meat of the procedure.  See below for an explanation.
  v_rows := DBMS_XMLStore.insertXML(
                                v_context,
                                XMLType.transform(p_xml_in, p_xsl_in));
  -- Close the context
  DBMS_XMLStore.closeContext(v_context);

END;
/
```

Ok. We have a procedure with 2 XML inputs and a VARCHAR2 input. p_xml_in is the external XML document. p_xsl_in is the transformation document. p_table is the table we want to insert into.

The XML procedures that came as an add-on to Oracle 8i did not require a context. Starting in 9i they did. It annoys me but I assume they have a reason for adding that. The context is a kind of object that holds information that the internal code can use. In this case it holds at least the table name. It may hold encoding information and other such info. For now, let's just say that you must use the context.

This line:

```
  v_rows := DBMS_XMLStore.insertXML(
                              v_context,
                              XMLType.transform(p_xml_in, p_xsl_in));
```

is the important one. I'll work from the back towards the front.

XMLType.transform(p_xml_in, p_xsl_in): This line applies the defined XSL to the input XML and returns a transformed document. If the XML and XSL were stored in a table, you could call XMLTransform from the select statement and accomplish exactly the same thing. This line is the one creating our canonical format.

The DBMS_XMLStore.insertXML call is the one that will actually parse our XML and write it to our table. In 9i, you may have used DBMS_XMLSave. DBMS_XMLSave is a wrapper for a Java routine that uses the DOM parser to save data to the database. In 10g, DBMS_XMLStore is the preferred package. DBMS_XMLStore is written in C and linked into the Oracle kernel. It also uses the SAX parser. For those reasons, DBMS_XMLStore has better performance characteristics.

On the last executable line of our procedure, we close that useful context.

Can you believe it? We are just about complete. Now all we have to do is call our procedure and pass it some XML.

I'm not going to repeat the XML or XSL text.

```
DECLARE
  -- Place the XML from above, in quotes in the XMLType() call
  v_xml XMLType := XMLType( XML Document from above );

  -- Place the XSL from above, in quotes in the XMLType() call
  v_xsl XMLType := XMLType( XSL Document from above );
BEGIN
  xml2tab(v_xml, v_xsl, 'tran_data');
END;
/
```

Badabing! Run that and then check your tran_data table for a record. If you have trouble getting the XML or XSL right, post a comment and I'll email a text file with the complete script. You don't need to include your email address in the comment but make sure you enter the correct one (where it asks for it) when you post your comment. That email address will not be displayed.

To me, that is a very simple and very dynamic method for getting a subset of an XML document into your database. No columns were hardcoded. In addition, if the application is designed right, even the table name can be dynamic.

If I were designing the application, I would put the XSL and the table name in a table. That way if there are multiple tables and/or XML types, you search on a type name (maybe embedded in the XML doc), return the table name and XSL and pass it to the procedure.

So, what would be a second way to insert some data?

```
CREATE OR REPLACE PROCEDURE xml2tab2(
                      p_xml_in XMLType,
                      p_table IN VARCHAR2 ) AS
  v_context DBMS_XMLStore.ctxType;
  v_rows NUMBER;

BEGIN
  v_context := DBMS_XMLStore.newContext(p_table);

  -- Change from the default ROW tag to the TRAN tag
  DBMS_XMLStore.setRowTag(v_context, 'TRAN');

  -- Identify which columns we want, ALL is the default
  DBMS_XMLStore.setUpdateColumn(v_context,'TRAN_ID');
  DBMS_XMLStore.setUpdateColumn(v_context,'RCODE');

  -- Do the insert
  v_rows := DBMS_XMLStore.insertXML(v_context, p_xml_in);

  DBMS_XMLStore.closeContext(v_context);

END;
/
```

In this procedure, first I create the context. Next, I override the default ROW tag with TRAN from our document. Remember that since we have a single row, ROWSET is optional. The next two lines are defining the columns that we want to insert. The XML document has four elements defined but we only want two of them. Without these two lines, the procedure would fail with an invalid column error. Finally, we call the insertXML procedure and close our context. Doesn't get much easier than that.

And to call the new procedure:

```
DECLARE
  -- Place the XML from above, in quotes in the XMLType() call
  v_xml XMLType := XMLType( XML Document from above );
BEGIN
  xml2tab2(v_xml, 'tran_data');
END;
/
```

Keeping it simple! If you only have a single table with a small set of columns, the second example is probably the way to go. If you need it dynamically at all or if you have a lot of tables, use the first example. If your XML is complex with nested elements and such, use the first example.

If you add or remove columns in the future, you would have to modify the code in example 2. You would only have to modify the XSL in example 1. And I hope you would store that XSL in a table and not code it into your program. Just stick it in an XMLType column.

And that's it for this example. One of the questions I hear most often is, "How do I get XML into the database?" This is just one way to answer a specific requirement.

I hope you get some use out of this.

XML and OO in Oracle, A Scenario

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006547.asp

Posted 11/17/2005

I'm battling the flu this week and have not been able to write much. I hope this entry will make sense to someone not enjoying a fever.

I was laying around thinking about XML and how my current project is passing the data back and forth. The application I'm designing is a service that will be called from at least two others applications and possibly more. The service is implemented via stored procedure that will be called by a C routine and by a JDBC call.

I'm often asked when and where to use OO in an Oracle application. One of my main goals in developing an application is maintenance. Most of the cost of a system is in the maintenance phase, not the development phase. That's where using OO as much as possible really does help. Not so much in the data model; I tend to stay relational there; but in the data structures used in the code (where it makes sense).

Ideally, I want to design a system to be both maintainable and performant. In some systems, real-time transaction processing for example, performance is the top goal. You might need to sacrifice a lot of maintainability to get the throughput you need. In my experience, OO in Oracle gives you a maintenance boost with some sacrifice in performance.

However, in most applications, you don't need to squeeze out the last bit of performance in a system. As long as it performs to expectations (and will scale as needed), you can sacrifice some performance for maintainability. In these applications, OO can be a great time saver both during design and later during maintenance.

My current application requires performance over maintainability. The transaction rate that I have to sustain doesn't give me much leeway to play. However, while I was thinking about the XML I'm using, a thought occurred that would make it easier for the clients to access their data.

I'm not using most of the features in XML DB for this application. The reasons are both political and performance related. In some ways that sucks, but in others it does force me to be creative.

Anyway, back to my thought. The XML that is getting passed around is fairly large (10s of k). Both clients and the database are interested in a few key fields. All data eventually gets logged but for processing only a subset is needed. I want a design that will allow each application to access the data it needs without bundling business logic in the application.

Here is our sample XML. We will use this XML throughout this entry.

```
<tran>
  <tran_id>1</tran_id>
  <tran_dt format="dd-mon-yyyy">01-JAN-2005</tran_dt>
  <tran_amt>100.00</tran_amt>
  <return_code>A</return_code>
  <ids>
    <id1>ABC</id1>
    <id2>XYZ</id2>
  </ids>
</tran>
```

For the clients to access tran_amt, they might walk a DOM tree or, if using Oracle native types, they would use a statement like this:

```
v_tran_amt := v_xml.extract('/tran/tran_amt/text()')
```

The client really doesn't care what the XML looks like and shouldn't need to know what the XML structure is. All it wants is the transaction amount. And remember, being a service, this XML is being passed through many sub-systems. Some of those systems use the data; some just pass it along. If the structure is hard coded into the application (via XPath), then anytime the structure of the XML changes, all of the dependant applications will need to be changed. In a perfect world, structures don't change; in my world, they change somewhat frequently.

That's where OO comes into it. I decided to create an object type that would give the clients what they want but would still be as robust and dynamic as XML.

Some additional info about the real XML document. There are hundreds of elements; some are nested. The users care primarily about a small subset of key fields. There is only a single, logical record. The root element does not contain multiple children.

For this example, I'm identifying tran_id, tran_dt and tran_amt as key fields. These key fields are the fields I want to be easily accessible.

So first, I create an object type:

```
CREATE OR REPLACE TYPE transaction_t AS OBJECT (
  XML_DATA xmltype,
  TRAN_ID number,
  TRAN_DT date,
  TRAN_AMT number,
  GOOD_RECORD char(1),
  ERROR_COND NUMBER,
  CONSTRUCTOR FUNCTION transaction_t( p_xml_data IN xmlType )
    RETURN SELF AS RESULT,
  CONSTRUCTOR FUNCTION transaction_t
    RETURN SELF AS RESULT,
  MEMBER PROCEDURE validate_record( p_xml_data IN XMLType );
)
/
```

Notice the GOOD_RECORD attribute. That will be populated based on the value of the return code. We will do a lookup against a return_codes table to populate this field. This is the return_codes table and a sample record:

```
CREATE TABLE return_codes (
  return_code CHAR(1) NOT NULL PRIMARY KEY,
  good_code CHAR(1) NOT NULL CONSTRAINT good_code_ck
       CHECK (good_code IN ('Y', 'N'))  -- Y means good, N means bad
  )
/

INSERT INTO return_codes (return_code, good_code)
  VALUES ('A', 'Y')
/
```

Now for the body of the type:

```
CREATE OR REPLACE TYPE BODY transaction_t AS
  CONSTRUCTOR FUNCTION transaction_t( p_xml_data IN xmlType )
    RETURN SELF AS RESULT IS
  BEGIN
    validate_record( p_xml_data );
    RETURN;
  END;

  CONSTRUCTOR FUNCTION transaction_t
    RETURN SELF AS RESULT IS
  BEGIN
    RETURN;
  END;

  MEMBER PROCEDURE validate_record( p_xml_data IN XMLType ) IS
  BEGIN
    SELF.tran_id := p_xml_data.extract('/tran/tran_id/text()').getNumberVal();
    SELF.tran_amt := p_xml_data.extract('/tran/tran_amt/text()').getNumberVal();
    SELF.tran_dt :=
             to_date(p_xml_data.extract('/tran/tran_dt/text()').getStringVal(),
                   p_xml_data.extract('/tran/tran_dt/@format').getStringVal() );

    BEGIN
      SELECT good_code
        INTO SELF.good_record
        FROM return_codes
        WHERE return_code =
              p_xml_data.extract('/tran/return_code/text()').getStringVal();
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
        SELF.good_record := 'N';
    END;

  EXCEPTION
    WHEN OTHERS THEN
      SELF.good_record := 'N';
      SELF.error_cond := SQLCODE;
  END;

END;
/
```

So, what are we doing here?

The first two functions are the constructors for the type. The first constructor handles initialization with allocation and the second does not. This is standard Oracle OOP.

The validate_record procedure is the one that encapsulates the business logic. This procedure is the one that serves up the key columns.

This procedure could be re-written as a GETTER function so that instead of initializing a set of attributes, it would retrieve an attribute in real-time. That would reduce the memory footprint but add a slight overhead at runtime. Depending on the use, it would be just as valuable. If you are going to dereference the same column many times, doing it once and keeping it in an attribute makes sense. That's how it is in my case.

A lot could be added:

- Update a field and have it update the xml
- Better exception handling
- Only allow type creation when using XML
- A print procedure for debugging
- Put the XPath Expression in a table and populate at object instantiation
- and more

Here is an anonymous block to test:

```
SET SERVEROUTPUT ON

DECLARE

  v_xml xmlType;

  v_myObject transaction_t;

BEGIN

   v_xml := XMLType(
       '<tran>
         <tran_id>1</tran_id>
         <tran_dt format="dd-mon-yyyy">01-JAN-2005</tran_dt>
         <tran_amt>100.00</tran_amt>
         <return_code>A</return_code>
         <ids>
           <id1>ABC</id1>
           <id2>XYZ</id2>
         </ids>
```

```
        </tran>');

  v_myObject := transaction_t(v_xml);

  dbms_output.put_line( 'Tran ID: ' || to_char( v_myObject.tran_id ) );

   v_xml := XMLType(
       '<tran>
        <tran_id>2</tran_id>
        <tran_dt format="dd-mon-yyyy">41-JAN-2005</tran_dt>
        <tran_amt>100.00</tran_amt>
        <return_code>A</return_code>
        <ids>
          <id1>ABC</id1>
          <id2>XYZ</id2>
        </ids>
        </tran>');

  v_myObject := transaction_t(v_xml);

  dbms_output.put_line( 'Tran ID: ' || to_char( v_myObject.tran_id ) );
  dbms_output.put_line( 'Error: ' || to_char( v_myObject.error_cond ) );

END;
/
```

Run that and see the results. The second XML doc results in an error. Do you see why?

In a way, this idea of mine duplicates some of what XML DB gives you out of the box. You could also do this using a package and some code. If you aren't concerned with persistence, I think this way nicely encapsulates the business knowledge (XPath Expressions) inside a single container. Your logic is shipped around with your data. Future maintenance is simplified.

IBM Viper: XML Technology in DB2

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006566.asp

Posted 11/18/2005

IBM announced a new version of DB2 called Viper. It looks like it's mostly a response to XML DB. It stores XML natively and you can access XML with SQL and XQuery. It also looks like it has optimizations and indexing for XML. Sounds a lot like the features Oracle has been adding since 9i. It even has Zend PHP integration.

You can go to the IBM site to get more info and join the beta. I plan to read the docs and see how Viper does compare to XML DB. I'll post what I find.

EDIT 10/18/05: This is not an open beta. I thought it was freely downloadable like Oracle 10g XE but you have to sign up and request to be included in the beta test. The request includes a questionnaire and if they like your answers, they will include you in the beta. I decided to pass for now.

Comments

Chris Eaton 11/18/2005

Hi Lewis, it may sound like 9i but it is actually quite different. In 10gr2 they describe an XMLType column as:

"XMLType tables and columns can be stored as Character Large Object (CLOB) values or as a set of objects. When stored as a set of objects, we refer to structured, or shredded storage."

The XML support in DB2 Viper stores the xml data as a hierarchical structure inside the relational database. It is not shredded or stored as a clob but stored truely in XML format. Many people call what they have a Native XML store so I think that meaning is missused enough to have little meaning. But there are significant performance and integrity advantages to keeping the XML in a hierarchical storage model compared to a clob or a shredded technique. I'll blog more about this soon on my blog so watch for more news coming soon.

Chris Eaton 11/18/2005

Here is a quote from Philip Howard, an independant analyst at Bloor Research.

"As far as the competition goes, Bloor said, IBM is leaving Oracle Corp. and Sybase Inc.—the database vendors with the "best current handle on XML"—well behind the curve.

"I expect to see Oracle, in particular, to froth at the mouth at this announcement," Bloor said.

The full story is here http://news.yahoo.com/s/zd/165544

LewisC 11/18/2005

Chris,

Thanks for the post. I look forward to reading about it!

I did read about the two different engines, one for XML and one for Relational. It sounds like a cool concept. I agree there is definitely overhead to shredding out to relational.

He loses credibility though when he says, "Blobs retain XML fidelity, but you lose the ability to search on data that's put into fields."

That's might currently be true in DB2, that's not true in Oracle. I can search and fully index any XML I want in Oracle without shredding. I even have a few different ways to do it.

However, having an actual hierarchical storage engine sounds pretty amazing, especially if it runs tightly with the relational engine. As far as comparing to Oracle though, XML DB is pretty amazing already.

I guess I'll have to wait to see though since the beta is not open to everyone. I also read today that IBM is planning on an "express edition" for DB2. I wonder if that will be viper?

Thanks for the post,

LewisC

Michael 11/18/2005

I refuted the Bloor Research Article over on my weblog some time ago. See

http://sqljunkies.com/WebLog/mrys/archive/2004/12/16/5599.aspx
http://sqljunkies.com/WebLog/mrys/archive/2005/04/20/12380.aspx

IBM is playing catch up here with Oracle 10gR2 and SQL Server 2005 w.r.t. their XML support. And all three provide native XML support (see the above April post for a reference to which IBM researchers contributed to).

Best regards
Michael

Michael,

LewisC 11/21/2005

Great articles and a great response to IBM! Thanks for the links.

LewisC

Chris Eaton 11/21/2005

Michael,

I don't really want to debate the term "Native" because I personally think the term is irrelevant. But I do have 2 comments on your blog posting

Based on your definition of Native XML store I think I could argue that a combination of Operating system files (to store the XML docs), and notepad (to update) and google desktop (to query) could define a native XML store. The point is that anyone can say they have a native xml store. But what is important are the attributes of storage, search, update, etc that customers can use to develop business applications (not the "marketing" names given to the features).

But more importantly you assert:
"What IBM's physical design is irrelevant. Whether you store it as a string, store it in some internal binary format making use of existing storage facilities provided by the relational database system or design a complete new storage engine does not matter."

I would disagree. And I think customers interested it the best performing system would also disagree. If DB2, Oracle or SQL Server stored integer datatypes as a character string inside the database and then every time you performed a sum opperation you had to first convert the characters to integers then I think your performance would be misserable and it would matter. I agree that customers should not care about internals but they do care about performance and it's up to the development engineers to find the optimal solution to deliver that performance. Shredding and storing as clobs does not give the best levels of performance for all types of operations. Shredding gives you good performance but is much less flexible (and one of the main benefits of XML is to allow for flexible schema evolution). Shredding also suffers when you want to reconstruct the original document (can lose digital signatures for example). CLOBs on the other hand are much more flexible and preserve the oringinal document. But performance suffers during search (indexes are required on EVERY search element otherwise you have to parse the entire clob at runtime...which is what the bloor article refered to when it said they were bad for searching...not that you can't search them but it is less efficient). As well updating portions of the clob are slower.

The hierarchical storage engine in DB2 viper delivers better performance then clobs without losing the flexibility.

Michael 11/21/2005

Hi Chris

The quote you give above is taken out of context. For the purpose of defining what a native store is, it does not matter how the physical layer looks like. Obviously, performance requirements will determine the appropriate physical implementation (which, in some cases may just be the blob/clob after all, if you just need to reserialize the whole document). Most likely, there will be different options provided using different indexing mechanism etc to provide the best trade-offs. And I am sure that IBM (as Oracle and Microsoft) are aware of that and will provide adequate mechanisms. Note that I refer to that in the Dec post and most of my articles on that topic.

As too saying that a IR search on your file system can be considered a native XML store: To some extend yes. Although note, that I talk about query in a much narrower and wider sense than search: It means structural query, filter and transformations: typical database stuff and none of that newfangled "googling" :-).

Best regards
Michael

Chris Eaton 11/22/2005

I think your statement on "trade-offs" is exactly the point I'm trying to make. If you have only CLOB or Shredded than you have to make a trade off.
CLOB = good flexibility but you have to trade off some level of performance (and the performance gets worse as the documents get larger). It's ok if you only want to retrieve the entire doc but subdocument update and subdocument retrieve pay a penalty.
Shred = good performance but you have to trade off flexibility. And the whole point behind the move people are making to XML is to have flexibility. Things change so when I am storing an XML document and the schema for that doc changes (think of goverment forms and how often they change :-) the shredding technique does not handle this very well.

That's why IBM went for a true hybrid model where XML data gets stored in a hierarchical format in the same way XML exists in the real world and relational data gets stored in rows and columns. This way you don't have to trade off performance or flexibility. In DB2 Viper an XML column can have zero, 1 or many schemas associated with it so as forms change, the new and old data can coexist. And you can get high performance search even for sublevel documents.

Michael Rys 11/27/2005

Chris,

I think I understand the hierarchical storage format that IBM is using. It is not that much different logically from a B-Tree. As such it is also decomposed, just in a more performant way than having to shred it into user-level tables. If you need to retrieve the full XML document in the same form as you put it in, you will see that it is slower than if you would have stored it in a LOB. So you will still have a trade-off, even with DB2.

Anyway, I am glad to see the investment that IBM is doing, since this will keep all of us busy out-innovating each other...

Chris Eaton 11/30/2005

It's very different than decomposition into user tables as that does not provide schema flexibility or document fidelity whereas a hierarchcial structure does. Yes it does not provide the same performance if all you want to do is extract the entire XML document but the beauty of a hybrid model is that you could also store the XML document in a CLOB in DB2 if all you ever wanted to do was store and retrieve (in fact you could store in both formats if you wanted to). In fact even Oracle can store the doc as a clob for fast store/retrieve. But I think the world is not just interested in that (otherwise we all would have stopped working on this stuff after we created CLOB datatypes).

Here is my attempt at trying to describe what each option (CLOB, Shred, Hierarchical) is bad, good or best at.

Hierarchical is represented by (H)
CLOB is represented by (C)
Shredded is represented by (S)

Information Fidelity (H)best (C)best (S)bad
Integration w/ relational (H)best (C)bad (S)best
Schema Felexibility (H)best (C)good (S)bad
Query / Update Performance (H)best (C)bad (S)good
Entire document retrieval (H)good (C)best (S)bad

Of course good, bad and best are relative but I think people can get the point.

LewisC 11/30/2005

Chris,

Thanks for the posts. Some questions/changes though (and giving viper performance the benefit of the doubt).

Information Fidelity (H)best (C)best (S)bad

Seems logical.

Integration w/ relational (H)best (C)bad (S)best

Why would hierarchical be the same as a relational (shredded)? Native relational has to have some advantages in joining with other relational objects. I think it would be: Integration w/ relational (H)good (C)bad (S)best

Schema Felexibility (H)best (C)good (S)bad

Why do you say hierarchical has better schema flexibility than CLOB? A CLOB is completely flexible. I say: Schema Felexibility (H)best (C)best (S)bad

Query / Update Performance (H)best (C)bad (S)good

My question here is the same as integration with relational. Are you saying that navigating a tree structure is better for query/update than relational? I find that hard to believe. I say: Query / Update Performance (H)Good (C)bad (S)Best

Entire document retrieval (H)good (C)best (S)bad

Agreed.

I can't wait to see some unbiased, third party performance metrics comparing Viper, 10gR2 and Sql Server.

I wonder if any of the open source databases can compete in this space?

Thanks,

LewisC

Chris Eaton 11/30/2005

Good discussion!

On integration with Relation, it would be true that the integration would be suboptimal if you were to compare with a standalone hierarchical XML database. But in the DB2 Viper case their is a hybrid where the XML at the data manager layer is stored hierarchically but everything above that layer is common so you can join that data with relational using SQL and/or XQuery just as efficiently.

On schema flexibility I should have explained it a bit more. Yes I completely agree that the schema is completely flexible in a CLOB store. However, only the application accessing that data can "unlock" the schema. The database has no smarts and so can't do much with the schema of the xml object. So perhaps it would be best to break this down into two comparisons (or give clob a good+ :-)

On query / update performance. Yes the shredding to relational allows for a good amount of scalability and some performance but it suffers from the fact that the mapping from XML to relational forces costly joins (in some cases this can be very costly) where as a hierarchical storage model does not require joining of mapped tables.

On the 3rd party performance metrics, I agree I'd like to see it. Alas it may never happen because both Oracle and MS license terms and conditions prevent 3rd parties from publishing benchmark results without Oracle/MS prior consent.

On open source there are several open source databases doing stuff. Here is a good link to XML database "stuff". http://www.rpbourret.com/xml/XMLDatabaseProds.htm

LewisC 12/1/2005

Chris,

Hmm. I guess I'll have to see for myself when IBM releases that free database! Have you heard any timeframes on that (or what the limitations might be)?

I've never understood why vendors wouldn't allow open benchmarking. Especially when they have such a great product. Anyway, I'm sure there will be some comparisons at some point.

And thanks for the excellent link! I've never run across that page. How that's possible eludes me but it'll take me a week just to scan all the links.

I noticed mysql isn't included at all under the XML-Enabled database link and postgreSQL is, but really isn't. Everything else is commercial. Sounds like a project for someone. heh EnterpriseDB may be adding SQLX support.

Thanks again,

LewisC

Binary XML - Compressing, Encrypting and Encoding Data in Oracle

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006917.asp

Posted 11/28/2005

In this entry, I'm going to cover how to encode, compress and encrypt a text string for inclusion in an XML document. XML by its definition is a human readable, text format. It is a series of fields (elements) and the textual values of those fields. Sometimes, though, you need to move binary data; things like pictures, digital signatures, or binary documents. At times, you may have data that you want to compress or encrypt before including it.

So how do you send binary data using XML? An emerging standard called XOP has been defined by the W3C. XOP is an acronym for XML-binary Optimized Packaging.

This is a quote from the W3C Proposed Recommendation:

A XOP package is created by placing a serialization of the XML Infoset inside of an extensible packaging format (such a MIME Multipart/Related, see [RFC 2387]). Then, selected portions of its content that are base64-encoded binary data are extracted and re-encoded (i.e., the data is decoded from base64) and placed into the package. The locations of those selected portions are marked in the XML with a special element that links to the packaged data using URIs.

In a number of important XOP applications, binary data need never be encoded in base64 form. If the data to be included is already available as a binary octet stream, then either an application or other software acting on its behalf can directly copy that data into a XOP package, at the same time preparing suitable linking elements for use in the root part; when parsing a XOP package, the binary data can be made available directly to applications, or, if appropriate, the base64 binary character representation can be computed from the binary data.

Before I show you an example script, let's talk a little about what encoding is. As I said above, sometimes you want to embed binary data in XML. You might also want to embed binary data in an email message. If you've ever heard of or used MIME (Multi-purpose Internet Mail Extension) to send a picture through email, you have used BASE64 encoding. Most modern email packages will automatically handle the encoding. I remember the days when I had to do it myself.

Anyway, BASE64 is the defacto standard for encoding. It basically takes the entire ASCII character set and represents it via ASCII letters. For a good explanation, I turn to the wikipedia definition for base64:

Base 64 literally means a positional numbering system using a base of 64. It is the largest power of two base that can be represented using only printable ASCII characters. This has led to its use as a transfer encoding for email among other things. All well-known variants of base 64 use the characters A-Z, a-z, and 0-9 in that order for the first 62 digits but the symbols chosen for the last two digits vary considerably between different systems that use base 64.

Now that is a concise definition!

Oracle provides support for both BASE64 and MIME encoding. I won't include MIME in this discussion. XOP does (or can) use it but the syntax in not that difficult and would just confuse the issue in this discussion. Personally, I don't like MIME. That invisible box thing gets on my nerves. heh Mime, Invisible box, get it? ;-)

Anyway, today I am going to use several of Oracle's supplied packages to encode, compress and encrypt a string. Since I will have an example of encoding a string (RAW actually), I will not encode the encrypted and compressed strings. I will leave that as an exercise for the reader. To do that, just do what I do for the first RAW buffer.

Enough explanation, here's the sample script:

```
DECLARE

  vBuffer VARCHAR2(2000);  -- Varchar text

  rBuffer RAW(32767);  -- RAW Buffer

  eBuffer RAW(32767);  -- Encoded Buffer

  cBuffer RAW(32767);  -- Encrypted Buffer

  tBuffer VARCHAR2(2000);  -- Text Encoded buffer

  crBuffer RAW(32767);  -- Compressed RAW buffer

  ceBuffer RAW(32767);  -- Compressed Encoded buffer

  v_key_length NUMBER := 256/8; -- encryption key length
  v_encryption_key RAW (32);
  v_encryption_type PLS_INTEGER := DBMS_CRYPTO.ENCRYPT_AES256
                               + DBMS_CRYPTO.CHAIN_CBC
                               + DBMS_CRYPTO.PAD_PKCS5;
BEGIN

  -- Initialize a string
  vBuffer := 'Hello. This is a test.' || chr(9) ||
             'If this had been actual data, ' || chr(13) ||
             'it would probably be in a table ' ||
             'somewhere.';

  -- Convert the string to RAW
  rBuffer := UTL_I18N.STRING_TO_RAW( vBuffer,  'AL32UTF8' );

  -- Encode RAW to base64
  eBuffer := UTL_ENCODE.base64_encode( rBuffer );

  -- Compress the RAW Buffer
  crBuffer := UTL_COMPRESS.lz_compress( rBuffer );

  -- Compress the encoded Buffer
  ceBuffer := UTL_COMPRESS.lz_compress( eBuffer );

  -- Encrpt the RAW Buffer
   v_encryption_key := DBMS_CRYPTO.randombytes(v_key_length);
   cBuffer := DBMS_CRYPTO.encrypt( rBuffer, v_encryption_type, v_encryption_key );

  -- Text Encode the string
  tBuffer := UTL_ENCODE.text_encode( vBuffer );

  -- What do we have in our varchar2 buffer?
  DBMS_OUTPUT.PUT_LINE( substr('vBuffer length: ' || to_char(length( vBuffer )) ||
                               ', vBuffer: ' || vBuffer
                        ,1, 200) );

  -- What do we have in our RAW buffer?
  DBMS_OUTPUT.PUT_LINE( substr('rBuffer length ' || to_char( length(rBuffer)) ||
                               ', rBuffer: ' || rBuffer
                        ,1, 200) );

  -- What do we have in our encoded buffer?
  DBMS_OUTPUT.PUT_LINE( substr('eBuffer length ' || to_char( length(eBuffer)) ||
                               ', eBuffer: ' || eBuffer
                        ,1, 200) );

  -- What do we have in our compressed raw buffer?
  DBMS_OUTPUT.PUT_LINE( substr('crBuffer length ' || to_char( length(crBuffer)) ||
                               ', crBuffer: ' || crBuffer
                        ,1, 200) );

  -- What do we have in our compressed encoded buffer?
  DBMS_OUTPUT.PUT_LINE( substr('ceBuffer length ' || to_char( length(ceBuffer)) ||
                               ', ceBuffer: ' || ceBuffer
                        ,1, 200) );

  -- What do we have in our encrypted buffer?
  DBMS_OUTPUT.PUT_LINE( substr('cBuffer length ' || to_char( length(cBuffer)) ||
                               ', cBuffer: ' || cBuffer
                        ,1, 200) );

  -- What do we have in our encoded buffer?
  DBMS_OUTPUT.PUT_LINE( substr('tBuffer length: ' || to_char(length( tBuffer )) ||
                               ', tBuffer: ' || tbuffer
                        ,1, 200) );

END;
/
```

Let's talk about this line-by-line:

```
 vBuffer VARCHAR2(2000);  -- Varchar text
```

A buffer to hold my string.

```
 rBuffer RAW(32767);  -- RAW Buffer
```

Most of these functions work on a datatype called a RAW. A RAW stores data that is uninterpreted by the Oracle engine.

```
  eBuffer RAW(32767);  -- Encoded Buffer
```

The buffer that I will use to store my BASE64 encoded string.

```
  cBuffer RAW(32767);  -- Encrypted Buffer
```

The buffer that I will use to store my DBMS_CRYPTO encrypted string.

```
  tBuffer VARCHAR2(2000);  -- Text Encoded buffer
```

Text encoding leaves text in a readable format but converts binary data (special characters like line feeds and tab characters) into viewable data. You'll see this in a web based URL most often. I include it just for comparison.

```
  crBuffer RAW(32767);  -- Compressed RAW buffer
```

The buffer that I will use to store my compressed raw buffer (without encoding).

```
  ceBuffer RAW(32767);  -- Compressed Encoded buffer
```

The buffer that I will use to store my compressed raw buffer (with encoding). Normally, you would compress a string and then encode it. I wanted to show what encoding does to string size, so I reversed that.

```
  v_key_length NUMBER := 256/8; -- encryption key length
  v_encryption_key RAW (32);
  v_encryption_type PLS_INTEGER := DBMS_CRYPTO.ENCRYPT_AES256
                               + DBMS_CRYPTO.CHAIN_CBC
                               •   DBMS_CRYPTO.PAD_PKCS5;
```

The above three variable are used by DBMS_CRYPTO to encrypt a string.

```
  -- Initialize a string
  vBuffer := 'Hello. This is a test.' || chr(9) ||
             'If this had been actual data, ' || chr(13) ||
             'it would probably be in a table ' ||
             'somewhere.';
```

This is the string that all of the following procedures use as a basis. Notice the CHR(9) which is a TAB special character and the CHR(13) which is a line feed.

```
  -- Convert the string to RAW
  rBuffer := UTL_I18N.STRING_TO_RAW( vBuffer,  'AL32UTF8' );
```

Here is convert the TEXT string to a RAW. You can use UTL_RAW.cast_to_raw to convert but the encryption routine expects a common NLS character set so I use this RAW conversion for all of the functions.

```
  -- Encode RAW to base64
  eBuffer := UTL_ENCODE.base64_encode( rBuffer );
```

I'm using base64_encode to encode the RAW buffer to BASE64.

```
  -- Compress the RAW Buffer
  crBuffer := UTL_COMPRESS.lz_compress( rBuffer );
```

Here, I'm using lz_compress to compress the unencoded RAW buffer. You will normally use lz_compress with BLOB or CLOB data. In this case, I only had a small string. When we look at the data below, we will see that compression works best on large data sets.

```
  -- Compress the encoded Buffer
  ceBuffer := UTL_COMPRESS.lz_compress( eBuffer );
```

Here, I'm using lz_compress to compress the encoded RAW buffer. Normally, you would compress the data and then encode it. I wanted to show what compression can do to a small string though.

```
  -- Encrypt the RAW Buffer
   v_encryption_key := DBMS_CRYPTO.randombytes(v_key_length);
   cBuffer := DBMS_CRYPTO.encrypt( rBuffer, v_encryption_type, v_encryption_key );
```

And here, I'm using DBMS_CRYPTO to encrypt my string. DBMS_CRPYTO has many options; I'm just displaying one of those.

```
  -- Text Encode the string
  tBuffer := UTL_ENCODE.text_encode( vBuffer );
```

And finally, the last of the encodings. This is the text encoding like you would see in a URL.

The rest of the script displays the size of the buffer and the value of the buffer. When you run it you should see something like this (I've truncated the lines to save space and make it more readable):

```
vBuffer length: 96, vBuffer: Hello. This is a test.     If this had been actual
rBuffer length 192, rBuffer: 48656C6C6F2E20546869732069732061207 4
eBuffer length 264, eBuffer: 53475673624738754946526F61584D6761584D6759
```

```
crBuffer length 206, crBuffer: 1F8B080000000000000B158B410E80200C04CF1E
ceBuffer length 284, ceBuffer: 1F8B080000000000000B1DCCBB0E83201400D0D
cBuffer length 224, cBuffer: 9ADFCB1FB5CF38A862C646A96FE2BEE2205F01C
tBuffer length: 103, tBuffer: Hello. This is a test.=09If this had been actual
data, =0Dit
```

And that's about it for generating binary data and encoding it for XML. Oracle provides everything you need to package it up quite nicely.

XE, XML and WebDAV - Access your XML data in Oracle XE

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006917.asp

Posted 12/14/2005

The question I probably receive most is how to create XML so that it can be accessed via ftp, http or even from the desktop and tools like MS Word. XML DB can do this and it's pretty easy to use. I'm going to use XE as my database of choice but this would work in any 10gR2 database. XE comes pre-configured to work, though. I'm going to do my best to keep this as simple as possible.

You may need to unlock the HR account if you haven't already done so. I will assume that the HR password is also HR, if not, adjust accordingly. I will be using the HR user and schema for my examples. Actually, I will be taking a couple of items directly from the Oracle XML DB documentation.

In SQL*Plus, run this DDL:

```
BEGIN
  DBMS_XMLSCHEMA.registerSchema('http://www.oracle.com/emp_simple.xsd',
    '<schema xmlns="http://www.w3.org/2001/XMLSchema"
             targetNamespace="http://www.oracle.com/emp_simple.xsd" version="1.0"
             xmlns:xdb="http://xmlns.oracle.com/xdb"
             elementFormDefault="qualified">
       <element name = "Employee">
         <complexType>
           <sequence>
             <element name = "EmployeeId" type = "positiveInteger"/>
             <element name = "Name" type = "string"/>
             <element name = "Job" type = "string"/>
             <element name = "Manager" type = "positiveInteger"/>
             <element name = "HireDate" type = "date"/>
             <element name = "Salary" type = "positiveInteger"/>
             <element name = "Commission" type = "positiveInteger"/>
             <element name = "Dept">
               <complexType>
                 <sequence>
                   <element name = "DeptNo" type = "positiveInteger" />
                   <element name = "DeptName" type = "string"/>
                   <element name = "Location" type = "positiveInteger"/>
                 </sequence>
               </complexType>
             </element>
           </sequence>
         </complexType>
       </element>
```

```
     </schema>',
    TRUE,
    TRUE,
    FALSE);
END;
/
```

And this one:

```
CREATE OR REPLACE VIEW emp_simple_xml OF XMLType
  XMLSCHEMA "http://www.oracle.com/emp_simple.xsd" ELEMENT "Employee"
   WITH OBJECT ID (extract(OBJECT_VALUE,
                          '/Employee/EmployeeId/text()').getnumberval()) AS
   SELECT
     XMLElement("Employee",
                XMLAttributes(
                  'http://www.oracle.com/emp_simple.xsd' AS "xmlns",
                  'http://www.w3.org/2001/XMLSchema-instance' AS "xmlns:xsi",
                  'http://www.oracle.com/emp_simple.xsd
                   http://www.oracle.com/emp_simple.xsd'
                  AS "xsi:schemaLocation"),
                XMLForest(e.employee_id     AS "EmployeeId",
                          e.last_name       AS "Name",
                          e.job_id          AS "Job",
                          e.manager_id      AS "Manager",
                          e.hire_date       AS "HireDate",
                          e.salary          AS "Salary",
                          e.commission_pct  AS "Commission",
                          XMLForest(
                            d.department_id     AS "DeptNo",
                            d.department_name  AS "DeptName",
                            d.location_id      AS "Location") AS "Dept"))
     FROM employees e, departments d
     WHERE e.department_id = d.department_id;
```

These two items are from the documentation and are what I will use for this example.

The first DDL above, registers an XML Schema with the database. XML Schemas are a topic unto themselves and if you would like more discussion on them, let me know with a comment below.

The second DDL is just creating an XMLType view based on the relational data in the HR.Employees and HR.Departments tables. It's using SQLX (XMLElement, XMLForest) to create the XMLType. I've talked about SQLX in a few of my past XML posts but if you want to see more, here's a good starter article: XML in the Database: A Brief Overview.

In SQL*PLus, run the following DDL:

```
DECLARE
  v_xml_dir VARCHAR2(30) := '/public/testXMLDir';
BEGIN

  IF dbms_xdb.CreateFolder(v_xml_dir) THEN
    dbms_xdb.setAcl(v_xml_dir,'/sys/acls/bootstrap_acl.xml');
    dbms_xdb.setAcl(v_xml_dir,'/sys/acls/all_owner_acl.xml');
    dbms_xdb.setAcl(v_xml_dir,'/sys/acls/ro_all_acl.xml');
    dbms_output.put_line('Success');
  ELSE
    dbms_output.put_line('FAILED!');
  END IF;

  COMMIT;

END;
/
```

What did we do here? v_xml_dir defines the name of a directory. dbms_xdb.CreateFolder creates that directory. /public is a directory that comes with XML DB.

The calls to dbms_xdb.setAcl set permissions on the folder. ACLs are another item that need an entry to themselves. ACL stands for Access Control List and is the way you protect data in XDB. Rather than get into the ACL stuff now, if you would like me to post an entry on it, leave me a comment below. The short story here is that HR owns the directory and can write to it and others can see it.

You don't really need the testXMLDir to do the steps below, you could do them in /public, but I figured I would try to keep your /public directory clean.

Now that we've defined this directory, let's navigate to the directory. We'll do that with an internet browser and then with the file explorer.

First, open up your browser of choice. I have tried this with IE and Firefox and would assume it will work with almost any other browser.

For the purposes of this test, I'm assuming that you have not locked down your database anymore than what comes installed and that you're running XE locally. If that's not true in your case, you might have to make some changes. If you have questions or problems, feel free to post them.

In your browser, type: **http://127.0.0.1:8080**

127.0.0.1 is your local loopback IP address and 8080 is the port that XML DB is listening on.

You should see something like:

```
Index of /

Name            Last modified                      Size
i/              Mon, 31 Oct 2005 16:28:31 GMT      -
public/         Mon, 05 Dec 2005 12:53:15 GMT      -
sys/            Mon, 10 Oct 2005 20:28:54 GMT      -
```

You're now browsing the database. Double click on the public/ link. You should see your new testXMLDir directory. Double click on that link to navigate to that directory. Sometimes it might take a minute or two for the directory to show up. Hit refresh a few times if you need to.

Now that we know everything is working as planned, open Explorer. Go to My Network Places and run the Add Network Place wizard. On the first screen, just hit Next.

Highlight "Choose another network location" and hit next.

In the "Internet or network address" field, enter **http://127.0.0.1:8080**

Look familiar? Hopefully it does. They say the memory is the 5th or 6th thing to go. heh

You should get a connect dialog asking for a user name and password. Enter HR/HR and select OK. Give it a name or accept the default. Hit next.

Make sure the "open this network place" is checked and hit finish. You should now see the new folder in your network places. Double click on that and then double click on the public folder. It will probably ask you to enter a user name and password again. Use HR/HR again.

You should see your /public/testXMLDir directory. Now grab a word document or something off your computer and drag it to this directory. See how easy that is? You can use this feature to share files. This is called WebDAV. You just stored a document from your PC in your database. Kewel!

Anyway, the point here was to make XML available. Remember that schema and view we created way up above?

Do a "DESC emp_simple_xml" in SQL*Plus to see the column defined in the view.

Now, run this script in SQL*Plus:

```
DECLARE

  CURSOR c_xmldocs IS
    SELECT sys_nc_rowinfo$ xmlData
      FROM emp_simple_xml x;

  v_res BOOLEAN;

BEGIN

  FOR xd IN c_xmldocs LOOP

    v_res := dbms_xdb.createResource(
             '/public/testXMLDir/'||
             xd.xmlData.extract(
                '/Employee/Name/text()').getStringVal() ||
             '_' ||
             xd.xmlData.extract(
                '/Employee/EmployeeId/text()').getStringVal() ||
                '.xml', xd.xmlData);

  END LOOP;

  COMMIT;

END;
/
```

Hit refresh (F5) in explorer. You should now see a bunch of XML files.

Ok, so what did this script do? The cursor grabbed all of the XML documents from our view. We looped through those and created a resource (i.e. a file) in our testXMLDir directory. The files are named Employee Name concatenated with an under bar (_) concatenated with the Employee's ID. Since XDB is transactional, we issue a commit.

In Explorer, double click on one of the XML documents, or highlight the XML document and choose open with and choose the application of your choice. You just made data in an XMLType view available from your desktop.

There are other ways to do this and there are other things you can do with this. Something in particular to look at is the DBURIType for this kind of data. Very powerful stuff.

You can also store XML here and then access it via SQL.

If you have any questions or issues, please let me know. I actually clarified a few things in my mind while writing this. Your questions will help me as well as yourself and others who are reading.

Comments

Peter Staunton 12/16/2005

LewisC,

thanks for the blog entry - very interesting stuff.

I've been going through the steps above with XE, and had one problem:

Logged in as HR, I try to execute this command...
dbms_xdb.setAcl(v_xml_dir,'/sys/acls/bootstrap_acl.xml');

. and get the following error.
ORA-31050: Access denied
ORA-06512: at "XDB.DBMS_XDB", line 91
ORA-06512: at line 13

As a result, I can't write to the test directory through Explorer:
"An error occurred copying all or some of the selected files."

I can, however, write to the Public directory at the higher level.

Is there some privilege that needs to be granted to HR so that he can execute DBMS_XDB.setAcl?

Thanks, Peter

Peter Staunton 12/16/2005
Just to follow up on my own question...

I logged in as SYS and granted XDBADMIN to HR, and then I was able to call dbms_xdb.setAcl as HR.

Problem solved.

When New Orleans was the Capita
of the Spanish Province of Lui-
siana.
1762 - 1803
This square bore the name
PLAZA D ARMAS

Jackson Square, New Orleans, LA – Christmas Day 2005

Chapter 7 - Objects

The great thing about Object Oriented code is that it can make small, simple problems look like large, complex ones.

The Sorcerer of OO, Part 1

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004077.asp

Posted 5/5/2005

I could have called this OO In Oracle: A Techie's View. But the Sorcerer of OO is a much cooler title. ;-) You have to pronounce OO as ooh! as in Ooh and Aah!

What is OO? OO stands for Object Oriented. OOD is OO Design, OOA is OO Analysis and OOP is Object Oriented Programming. Oracle is an ORDBMS, an Object Relational Database Management System. In this article, I am going to present a sample java class and show how you would implement the same class in Oracle. This is a technical article comparing Oracle objects with Java. In the next couple of articles, I will demonstrate inheritance, polymorphism and other traits expected of an OO language. I will also cover type evolution, an important aspect of types in Oracle.

PL/SQL, the procedural language for Oracle, is not particularly OO in the classical sense. It does provide quite a bit of OOness though. Via packages, you have encapsulation and abstraction. PL/SQL is not really intended to be OO but object methods are implemented using PL/SQL.

Oracle provides Object Types to implement the OO in ORDBMS. One of the strengths of an RDBMS is the abundance of generic data types. A type rich environment lets you express your data naturally. Object Types let you define more specific types using the existing types in the system.

Let's create a class and a demo in Java and then we'll create the equivalent class and demo in PL/SQL.

```
public class Calculator
{
  double value;
  public Calculator( double amt )
  {
    value = amt;
    print();
  }
  public Calculator( )
  {
    value = 0;
    print();
  }
  public void add( double amt )
  {
    value +=  amt;
    print();
  }
  public void subtract( double amt )
  {
    value -= amt;
    print();
  }
  public void print()
  {
    System.out.println( "Value is: " + value );
  }
}
```

What this class does is define a Calculator type. The type has one attribute: value and three methods: add, subtract and print. There are also two constructors (initializing functions), one with a parameter and one with out.

Here is a test class to call this class:

```
class demo
{
  public static void main(String args[])
  {
    Calculator calc = new Calculator(1);
    Calculator calc1 = new Calculator();
    calc.add(5);
    calc1.add(10);
    calc1.subtract(2);
  }
}
```

It's pretty easy to read. "class demo" is the name of our demo, " public static void main(String args[])" tells Java that this will be a procedure to run rather than be called from another class. We next create a new calc variable and initialize its value to 1. We then create a variable called calc1 and let it value default to 0 (see the constructor in the class). Then we add 5 to variable calc, add 10 to variable calc1 and finally subtract 2 from calc1.

If you run this, you will get:

```
Value is 1.0
Value is 0.0
Value is 6.0
Value is 10.0
Value is 8.0
```

And now for the equivalent in Oracle:

In Oracle, creating a "class" is a two-step process. First, we have to declare the object type:

```
CREATE OR REPLACE TYPE Calculator AS OBJECT (

  value NUMBER,

  CONSTRUCTOR FUNCTION Calculator( value IN NUMBER )
         RETURN SELF AS RESULT,

  CONSTRUCTOR FUNCTION Calculator
         RETURN SELF AS RESULT,
  MEMBER PROCEDURE print,

  MEMBER PROCEDURE add1( p_amt IN NUMBER ),

  MEMBER PROCEDURE subtract( p_amt IN NUMBER )

  )

  NOT FINAL;

/
```

Here we are creating a type with one attribute: value and three methods: add1, subtract and print.

Note that add is apparently a reserved word so I added the 1 to add to get add1. Subtract is not a reserved word.

As in the java class, we created two constructors; one with a parameter and one without.

The NOT FINAL is a directive that will allow us to SUBTYPE this TYPE. This is required for a subtype to inherit the attributes and methods of a supertype.

Did you notice that the type is just declarative? That is what I meant that a type in Oracle is a two-step process. Here is the executable portion of our type:

```
CREATE OR REPLACE TYPE BODY Calculator AS

  CONSTRUCTOR FUNCTION Calculator( value IN NUMBER )

     RETURN SELF AS RESULT IS

  BEGIN

    SELF.value := value;

    print;

    RETURN;

  END;

  CONSTRUCTOR FUNCTION Calculator

     RETURN SELF AS RESULT IS

  BEGIN

    SELF.value := 0;

    print;

    RETURN;

  END;

  MEMBER PROCEDURE print IS
  BEGIN

    DBMS_OUTPUT.PUT_LINE( TO_CHAR( value ) );

  END;

  MEMBER PROCEDURE add1( p_amt IN NUMBER )  IS

  BEGIN

    value := value + p_amt;

    print;

  END;

  MEMBER PROCEDURE subtract( p_amt IN NUMBER )  IS

  BEGIN

    value := value - p_amt;

    print;

  END;

END;

/
```

Constructors initialize a type. Before you can assign variables to or access a type, you MUST initialize it. Most types will let you assign values to the type as part of the initialization. The first constructor, the one with the parameter is assigning a 1 to the attribute value.

By default, Oracle creates a default constructor for every type you create. A constructor is ALWAYS the same name as the type and the default constructor takes as parameters the list of attributes of that type. So, for the example above, Oracle automatically created a Calculator(value IN NUMBER) constructor. I overrode that constructor so that I could make the call to print in that function. The constructor MUST have as parameters the exact list of attributes and the exact list of data types to override the default constructor. If I had called the parameter to my constructor, p_value instead of value, i.e. Calculator(p_value IN NUMBER), I would have received a runtime error (PLS-00307) when accessing it.

Every constructor and method is automatically assigned a parameter called SELF. SELF refers to the INSTANCE of the object. In the constructor, notice that I referred to the attribute as SELF.value and the parameter just as value. When you see SELF in an object, it is referring to the current instance of that object.

And now for the demo:

```
DECLARE

  calc Calculator;
  calc1 Calculator;

BEGIN

  calc := Calculator(1);

  calc1 := Calculator();

  calc.add1(5);

  calc1.add1(10);

  calc1.subtract(2);

END;

/
```

If you run this in sql*plus (you have to turn on serveroutput first), you will get:

```
Value is 1.0
Value is 0.0
Value is 6.0
Value is 10.0
Value is 8.0
```

And that does it. This is the basic format for creating a type in oracle. Like Java, it is very straightforward. I think it's even more readable than Java but I'm biased.

Next up will be inheritance.

The Sorcerer of OO, Part 2

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004151.asp

Posted 5/14/2005

This is a continuation of Object Orientation in Oracle. I guess I should have called it the Sorcerer of OOO. This is a technical entry.

Anyway, in part 1 I covered the basics of creating an Oracle object type and showing how it compared to a Java class. Today I'm going to cover the concepts of comparing objects and Inheritance. In part 3, I will cover type evolution and Polymorphism and in part 4 I will cover Object Views and Object Tables.

Comparing Objects

Many times, you will be working with multiple instances of the same object. You need a way to see if the instances, the instantiated object's values, are the same. Oracle provides the ability to compare object instances using MAP or ORDER methods. In the type definition, you can declare a MAP method or an ORDER method but not both.

A MAP member function uses a scalar value for comparison. This function returns either a number, character or date column that can directly be compared with another object. You could, for instance, compare a primary key.

Continuing on with the sample class we developed in part 1, let's modify our Calculator type and add a map member function. I'm going to drop the other methods for now to save space.

We drop the existing type:

```
DROP type calculator;
```

Create the new object:

```
CREATE OR REPLACE TYPE Calculator AS OBJECT (
  value NUMBER,
  MAP MEMBER FUNCTION comp RETURN NUMBER
  )
  NOT FINAL;
/
```

And the object Body:

```
CREATE OR REPLACE TYPE BODY Calculator AS
  MAP MEMBER FUNCTION comp RETURN NUMBER IS
  BEGIN
    RETURN value;
  END;
END;
/
```

Turn on serveroutput so that we can see the debug messages:

```
SET SERVEROUTPUT ON
```

And run the demo:

```
DECLARE
  calc Calculator;
  calc1 Calculator;
BEGIN

  -- Init 2 calculator with the same value and compare
  calc := Calculator(1);
  calc1 := Calculator(1);

  IF calc = calc1 THEN
    DBMS_OUTPUT.put_line( 'They are equal.' );
  ELSE
    DBMS_OUTPUT.put_line( 'They are not equal.' );
  END IF;

  -- Init 2 calculator with different values and compare
  calc := Calculator(1);
  calc1 := Calculator(2);

  IF calc = calc1 THEN
    DBMS_OUTPUT.put_line( 'They are equal.' );
  ELSE
    DBMS_OUTPUT.put_line( 'They are not equal.' );
  END IF;

END;
/
```

And that's how a MAP function works. In a more complex type, you may add up all of the numeric values and create a hash total or any other functionality as long as it returns a single scalar value.

But what if you need to compare more complex criteria. What if you need to compare every attribute in an instance against another instance? In this case, you would use an ORDER member function. An ORDER function accepts a single parameter, an instance of the type, and compares that instance against the current instance. An ORDER method must always return either -1, 0 or 1. Let's recreate our type and see how this works.

Drop the type and recreate with an Order Member Function:

```
DROP type calculator;
CREATE OR REPLACE TYPE Calculator AS OBJECT (
  value NUMBER,
  ORDER MEMBER FUNCTION comp(p_calc Calculator) RETURN INTEGER
  )
  NOT FINAL;
/
```

And the new body:

```
CREATE OR REPLACE TYPE BODY Calculator AS

  ORDER MEMBER FUNCTION comp( p_calc Calculator) RETURN INTEGER IS
    v_ret_val INTEGER;
  BEGIN
    CASE
    WHEN SELF.value < p_calc.value THEN
      v_ret_val := -1;
    WHEN SELF.value = p_calc.value THEN
      v_ret_val := 0;
    WHEN SELF.value > p_calc.value THEN
      v_ret_val := 1;
    END CASE;

    RETURN v_ret_val;
  END;

END;
/
```

And then run the demo:

```
SET SERVEROUTPUT ON

DECLARE
  calc Calculator;
  calc1 Calculator;
  calc2 Calculator;
BEGIN
  calc := Calculator(1);
  calc1 := Calculator(1);
  calc2 := Calculator(2);
  -- They are the same, expect 0
  DBMS_OUTPUT.put_line( 'Calc(calc1): ' ||
        to_char(calc.comp(calc1)) );

  -- Calc less than calc2, expect -1
  DBMS_OUTPUT.put_line( 'Calc(calc2): ' ||
        to_char(calc.comp(calc2)) );

  -- Calc2 greater than calc1, expect 1
  DBMS_OUTPUT.put_line( 'Calc2(calc1): ' ||
        to_char(calc2.comp(calc1)) );

END;
/
```

Obviously, this is a very simple example. This particular type is better suited for a MAP function rather than an ORDER function. An ORDER function is useful when you have a complex comparison needs and/or need to compare multiple attributes. You might decide that if a key attribute is null, it will be less than a completely full instance and the ORDER will return a -1. You might also decide that an instance with all attributes filled is always a greater value than a partially full instance. The rule I follow is how complex the COMPARISON of the type is, not necessarily the complexity of the type itself.

And that's type comparison in a nutshell. Let's move on to Inheritance.

Inheritance

What is type inheritance? A child might inherit a parent's eye or hair color. You could say that eye and hair color are attributes of both the parent and the child. Type inheritance works along the same lines.

Let's say this is our type:

```
CREATE OR REPLACE TYPE Calculator AS OBJECT (
  value NUMBER,
  CONSTRUCTOR FUNCTION Calculator( value IN NUMBER ) RETURN SELF AS RESULT,
  CONSTRUCTOR FUNCTION Calculator RETURN SELF AS RESULT,
  MAP MEMBER FUNCTION comp RETURN NUMBER,
  MEMBER PROCEDURE print,
  MEMBER PROCEDURE add1( p_amt IN NUMBER ),
  MEMBER PROCEDURE subtract( p_amt IN NUMBER )
  )
  NOT FINAL;
/
```

We also have our associated type body.

```
CREATE OR REPLACE TYPE BODY Calculator AS

  CONSTRUCTOR FUNCTION Calculator( value IN NUMBER )
     RETURN SELF AS RESULT IS
  BEGIN
    SELF.value := value;
    print;
    RETURN;
  END;
  CONSTRUCTOR FUNCTION Calculator
     RETURN SELF AS RESULT IS
  BEGIN
    SELF.value := 0;
    print;
    RETURN;
  END;

  MAP MEMBER FUNCTION comp RETURN NUMBER IS
  BEGIN
    RETURN value;
  END;

  MEMBER PROCEDURE print IS
  BEGIN
    DBMS_OUTPUT.PUT_LINE( TO_CHAR( value ) );
  END;

  MEMBER PROCEDURE add1( p_amt IN NUMBER )  IS
  BEGIN
    value := value + p_amt;
    print;
  END;

  MEMBER PROCEDURE subtract( p_amt IN NUMBER )  IS
  BEGIN
    value := value - p_amt;
    print;
  END;

END;
/
```

Now we want to create a more advanced version of our type. Let's call it the AdvancedCalculator. Our AdvancedCalculator has an additional attribute and two new methods. The new type will inherit all of the existing functionality of our base, or super type.

The last statement in our create type command is NOT FINAL. This allows us to subtype this type. If those keywords are not present, you cannot create a subtype.

```
CREATE TYPE AdvancedCalculator UNDER Calculator (
  pi FLOAT,
  CONSTRUCTOR FUNCTION AdvancedCalculator RETURN SELF AS RESULT,
  MEMBER PROCEDURE multiply( p_amt IN NUMBER ),
  MEMBER PROCEDURE divide( p_amt IN NUMBER )
   )
NOT FINAL;
/
```

If you describe the new AdvancedCalculator, you will see that everything available with Calculator is now available for AdvancedCalculator. I created a constructor for this type so that I won't have to supply Pi when I use it.

Now we will create our body:

```
CREATE OR REPLACE TYPE BODY AdvancedCalculator AS

  CONSTRUCTOR FUNCTION AdvancedCalculator
     RETURN SELF AS RESULT IS
  BEGIN

    SELF.value := 0;
    self.pi := 3.14159;
    SELF.print;

    RETURN;
  END;

  MEMBER PROCEDURE multiply( p_amt IN NUMBER )  IS
  BEGIN
    value := value * p_amt;
    SELF.print;
  END;

  MEMBER PROCEDURE divide( p_amt IN NUMBER )  IS
  BEGIN
    value := value / p_amt;
    SELF.print;
  END;

END;
/
```

Our AdvancedCalculator has inherited the VALUE attribute and all of the methods of Calculator. It added a new PI attribute, a new constructor and 2 new member procedures. Below is a sample program:

```
DECLARE
  calc Calculator;
  calc1 AdvancedCalculator;
BEGIN

  calc := Calculator(1);
  calc1 := AdvancedCalculator();

  calc.add1(5);
  calc1.add1(3);
  calc1.multiply(10);
  calc1.divide(2);

  IF calc != calc1 THEN
    DBMS_OUTPUT.PUT_LINE( 'Calc, ' ||
          to_Char(calc.value) || ', does not equal calc1, ' ||
          to_char(calc1.value) );
  END IF;

  calc.value := 5;
  calc1.value := 5;

  IF calc = calc1 THEN
```

```
    DBMS_OUTPUT.PUT_LINE( 'Calc, ' ||
          to_Char(calc.value) || ', does equal calc1, ' ||
          to_char(calc1.value) );
  END IF;

END;
/
```

Notice that we used Calculator interchangeably with AdvancedCalculator. That is where the power of inheritance really kicks in. Oracle is aware of the type hierarchy and knows how to compare them. It also knows when a type can be used in place of a supertype and vice versa. In part 4, I will also talk about the CAST function.

And here is an important note: The subtype does not copy the attributes and methods from the supertype. They are permanently linked. If you change the behavior or attributes of the supertype, the subtype will reflect those changes, i.e. it will inherit them. And if that's not scary enough, type can be implemented as physical structures in the database, i.e. table and views. More on that in the future though.

And that's it for today. In Part 3 I will cover type evolution and polymorphism. In Part 4 I will cover object views and object tables.

If you're interested, here's a few OO, not Oracle, links I've read recently:

David Foderick's Blog

Why Object Oriented Encapsulation is Obsolete

Teaching Object Oriented Programming

Intro to Object-Oriented Programming with Java

The Sorcerer of OO, Part 3

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004340.asp

Posted 5/26/2005

In Part 1 of this series, we talked about how Oracle objects compare to Java and how to create Oracle Objects. In Part 2, I covered we covered object comparison and type inheritance. Today, we'll talk about polymorphism and type evolution.

Like parts 1 and 2, this will be a technical discussion.

Polymorphism

First a definition. What is Polymorphism and why is it important? One of the best definitions I have found is this link at OnJava. You can skip the part about object serialization. Not really that important to Oracle's OO as you're already in the database.

Let's go back to our (most basic) calculator example.

```
CREATE OR REPLACE TYPE Calculator AS OBJECT (
  value NUMBER,
  MEMBER PROCEDURE print,
  MEMBER PROCEDURE add1( p_amt IN NUMBER ),
  MEMBER PROCEDURE subtract( p_amt IN NUMBER )
  )
  NOT FINAL;
/

CREATE OR REPLACE TYPE BODY Calculator AS
  MEMBER PROCEDURE print IS
  BEGIN
    DBMS_OUTPUT.PUT_LINE( TO_CHAR( value ) );
  END;

  MEMBER PROCEDURE add1( p_amt IN NUMBER )  IS
  BEGIN
    value := value + p_amt;
  END;

  MEMBER PROCEDURE subtract( p_amt IN NUMBER )  IS
  BEGIN
    value := value - p_amt;
  END;

END;
/
```

And our Advanced Calculator:

```
CREATE OR REPLACE TYPE AdvancedCalculator UNDER Calculator (
  pi FLOAT,
  CONSTRUCTOR FUNCTION AdvancedCalculator
     RETURN SELF AS RESULT ,
  MEMBER PROCEDURE multiply( p_amt IN NUMBER ),
  MEMBER PROCEDURE divide( p_amt IN NUMBER ),
  OVERRIDING MEMBER PROCEDURE print
   )
NOT FINAL;
/
```

This isn't exactly the same AdvancedCalculator. Notice the OVERRIDING keyword. I'm overriding the print method in the subtype AdvancedCalculator.

And let's create a new body for our subtype and change the print method to include data about the attribute pi:

```
CREATE OR REPLACE TYPE BODY AdvancedCalculator AS

  CONSTRUCTOR FUNCTION AdvancedCalculator
     RETURN SELF AS RESULT IS
  BEGIN

    SELF.value := 0;
    self.pi := 3.14159;

    RETURN;
  END;
  MEMBER PROCEDURE multiply( p_amt IN NUMBER )  IS
  BEGIN
    value := value * p_amt;
  END;

  MEMBER PROCEDURE divide( p_amt IN NUMBER )  IS
  BEGIN
    value := value / p_amt;
  END;

  OVERRIDING MEMBER PROCEDURE print IS
  BEGIN
    DBMS_OUTPUT.PUT_LINE( 'Value: ' || TO_CHAR( value )
                             || ', Pi: ' || TO_CHAR( pi ) );
  END;

END;
/
```

And a demo script:

```
DECLARE
  calc Calculator;
  calc1 AdvancedCalculator;
  calc2 AdvancedCalculator;
BEGIN
  calc := Calculator(1);
  calc1 := AdvancedCalculator();
  calc2 := AdvancedCalculator();

  calc.print;
  calc1.print;
  calc2.print;

  calc.add1(5);
  calc1.add1(5);
  calc2.add1(5);

  calc.print;
  calc1.print;
  calc2.print;

END;
/
```

At it's most basic, that is polymorphism. We defined a base type and some basic functionality. We then defined a subtype and while most of its functionality applied, we needed to make a minor change to the print method. We did this by OVERRIDING the print method.

Like most examples, this one is somewhat pointless except to describe how to implement polymorphism. If you look back at the link above, they mentioned the standard Shape example and calculating area. In a business process, you might have different types of transactions with different billing calculations. Polymorphism doesn't look useful until you need it.

This could have been implemented by implementing a new method in the subtype, print1 or printAdvanced or something along those lines but if you did that, you would defeat the whole purpose of using objects. You, the programmer, would have to keep track of each type and decide programmatically which kind of print method to call. In this example, that would be pretty easy but in the real world with thousands of lines of code and multiple sub-types, it would get messy.

I'm going to leave the definition of polymorphism here for now. In Part 4, I will show some of the power of this when I get into using object views.

Type Evolution

I wanted to talk about this because it has important aspects for maintenance of an application. In Oracle 8i, the first version I used OO in Oracle, you had to drop and recreate your types to make changes. What a PITA that was. It was particularly annoying when you used a type in an AQ queue and then wanted to change it. Arghh!

Well, in 10g, you can change a type within some limitations.

Using our calculator examples, let's change some items.

```
alter type advancedcalculator drop member procedure print
/
```

Note: If you get an ORA-21700, just disconnect and reconnect. I'm not sure why that happens but reconnecting fixes it for me. Metalink gives a few possibilities but it seems to be a caching issue for me.

When you alter the type, you have to recompile the body. Since we're dropping a method, we need to modify the code to not include the print method:

```
CREATE OR REPLACE TYPE BODY AdvancedCalculator AS

  CONSTRUCTOR FUNCTION AdvancedCalculator
     RETURN SELF AS RESULT IS
  BEGIN

    SELF.value := 0;
    self.pi := 3.14159;

    RETURN;
  END;

  MEMBER PROCEDURE multiply( p_amt IN NUMBER )  IS
  BEGIN
    value := value * p_amt;
  END;

  MEMBER PROCEDURE divide( p_amt IN NUMBER )  IS
  BEGIN
    value := value / p_amt;
  END;

END;
/
```

This removes the print procedure from the subtype. Do a desc on AdvancedCalculator.

Whoa! It's still there. But is it? Run our demo script from the polymorphism example above.

Remember, we had overridden the print method from its supertype. By removing it from the sub-type, it still exists in the supertype and is still available to the subtype.

Let's remove the subtract method from the supertype:

```
alter type calculator drop member procedure subtract( p_amt IN NUMBER )  CASCADE
/
```

Recompile the body without the subtract procedure:

```
CREATE OR REPLACE TYPE BODY Calculator AS
  MEMBER PROCEDURE print IS
  BEGIN
    DBMS_OUTPUT.PUT_LINE( TO_CHAR( value ) );
  END;

  MEMBER PROCEDURE add1( p_amt IN NUMBER )  IS
  BEGIN
    value := value + p_amt;
  END;

END;
/
```

Do a desc on advancedcalculator. On my system, I could still see the method subtract and had to disconnect and reconnect to get rid of it. It definitely looks like there are caching issues with types.

Anyway, the CASCADE option alters the subtype in addition to the supertype. By changing the supertype, the subtype evolved. Compared to 8i this is awesome.

Again, I'm going to stop here but in Part 4 I will cover object views and object tables and you will see in greater detail how polymorphism and type evolution works.

Just to clarify: Polymorphism is the ability of a subtype to change, or morph, its behavior without changing the supertype or other subtypes of the supertype and type evolution is the ability to evolve a type that has dependencies.

I tried to be brief but hopefully I wasn't so brief that this isn't useful. The two topics today are really intros to part 4 where we can finally get into the meat of OO. Object Views are useful in everyday development. Object Tables are less so but like I say, I'd rather have it available and not use it than need it and not have it.

Sorcerer of OO - Part 4

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/005014.asp

Posted 7/19/2005

In Part 1 of this series, we talked about how Oracle objects compare to Java and how to create Oracle objects. In Part 2, we covered object comparison and type inheritance. And in Part 3, we covered polymorphism and type evolution. Today, I'm covering the creation of Object Tables and Object Views.

Creating an object table is very simple. First, you create the types you need and then you use standard CREATE TABLE syntax. XMLType is an object type. We can use that to create an object table in two different ways; as a row object and as a column object.

Note: I'll be using the SCOTT demo account for all of the following commands.

Object Tables

Create a row object table:

```
CREATE TABLE myXMLType OF XMLType;
```

Insert some records:

```
INSERT INTO myXMLType
  (SELECT
        XMLElement("Employees",
        XMLForest( ename, empno ) )
     FROM EMP );
```

View the records:

```
SQL> l
  1* SELECT * from myxmltype
SQL> /
SYS_NC_ROWINFO$
-----------------------------------------------------------------
<Employees><ENAME>SMITH</ENAME><EMPNO>7369</EMPNO></Employees>
<Employees><ENAME>ALLEN</ENAME><EMPNO>7499</EMPNO></Employees>
<Employees><ENAME>WARD</ENAME><EMPNO>7521</EMPNO></Employees>
<Employees><ENAME>JONES</ENAME><EMPNO>7566</EMPNO></Employees>
<Employees><ENAME>MARTIN</ENAME><EMPNO>7654</EMPNO></Employees>
<Employees><ENAME>BLAKE</ENAME><EMPNO>7698</EMPNO></Employees>
<Employees><ENAME>CLARK</ENAME><EMPNO>7782</EMPNO></Employees>
<Employees><ENAME>SCOTT</ENAME><EMPNO>7788</EMPNO></Employees>
<Employees><ENAME>KING</ENAME><EMPNO>7839</EMPNO></Employees>
<Employees><ENAME>TURNER</ENAME><EMPNO>7844</EMPNO></Employees>
<Employees><ENAME>ADAMS</ENAME><EMPNO>7876</EMPNO></Employees>
<Employees><ENAME>JAMES</ENAME><EMPNO>7900</EMPNO></Employees>
<Employees><ENAME>FORD</ENAME><EMPNO>7902</EMPNO></Employees>
<Employees><ENAME>MILLER</ENAME><EMPNO>7934</EMPNO></Employees>
14 rows selected.
```

Now we do the same thing with a column object:

```
CREATE TABLE myXMLType2 (
  emp_no NUMBER(4) NOT NULL PRIMARY KEY,
  emp_info XMLType );
INSERT INTO myXMLType2 (emp_no, emp_info)
  (SELECT empno,
        XMLElement("Employees",
        XMLForest( ename, empno ) )
     FROM EMP );
SQL> l
  1* SELECT * from myxmltype2
```

```
SQL> /
    EMP_NO
----------
EMP_INFO
----------------------------------------------------------------
      7369
<Employees><ENAME>SMITH</ENAME><EMPNO>7369</EMPNO></Employees>

      7499
<Employees><ENAME>ALLEN</ENAME><EMPNO>7499</EMPNO></Employees>

      7521
<Employees><ENAME>WARD</ENAME><EMPNO>7521</EMPNO></Employees>

      7566
<Employees><ENAME>JONES</ENAME><EMPNO>7566</EMPNO></Employees>

      7654
<Employees><ENAME>MARTIN</ENAME><EMPNO>7654</EMPNO></Employees>
.
.
14 rows selected.
```

Updates and deletes are just as easy as inserts and selects.

You may hear about REFs and particularly, Dangling REFs. REFs are a way to link to an Object ID (OID). When you DEREF a REF, you link to the object instance with that OID.

In my opinion, REFs are a way around foreign keys and are a bad idea. Even though Oracle is an ORDBMS, I stick with the R and not the O when I need to join objects. I think REFs make maintenance more difficult than it needs to be. That's all I will say about REFs except use them at your own peril.

And that is object tables in a nutshell. The two examples above (using XMLTYPE specifically) are probably 95% of what I have ever used object tables for. I don't use object tables to store child tables inside of a parent table. I use object views to present data that way when needed.

Many examples I see are something like a Dept table with embedded employees, i.e.:

Create an Employee Type:

```
CREATE TYPE emp_info AS OBJECT (
  empno NUMBER(4),
  ename VARCHAR2(10),
  job VARCHAR2(9) );
/
```

Create a collection from the type. A collection, nested table or varray, can store multiple records.

```
CREATE TYPE emps AS TABLE OF emp_info;
/
```

Create the object table (in this case a column object table):

```
CREATE TABLE dept_obj (
  deptno NUMBER(2) NOT NULL PRIMARY KEY,
  dname VARCHAR2(14),
  loc VARCHAR2(13),
  emp_data emps )
NESTED TABLE emp_data STORE AS emps_d
/
```

Insert some rows. Notice the CAST(MULTISET(syntax. CAST tells oracle that the result is of object type emps and MULTISET tells Oracle that it will be a collection.

```
INSERT INTO dept_obj (deptno, dname, loc, emp_data)
  (  SELECT deptno, dname, loc,
       CAST( MULTISET (
         SELECT empno, ename, job
           FROM emp
           WHERE emp.deptno = dept.deptno)
           AS emps)
           AS emp_info
   FROM dept );
```

Select the data:

```
SQL> select * from dept_obj ;
    DEPTNO DNAME          LOC
---------- -------------- -------------
EMP_DATA(EMPNO, ENAME, JOB)
-----------------------------------------------------------------------------------
-------
        10 ACCOUNTING     NEW YORK
EMPS(EMP_INFO(7782, 'CLARK', 'MANAGER'), EMP_INFO(7839, 'KING', 'PRESIDENT'),
EMP_INFO(793
4, 'MILLER', 'CLERK'))

        20 RESEARCH       DALLAS
EMPS(EMP_INFO(7369, 'SMITH', 'CLERK'), EMP_INFO(7566, 'JONES', 'MANAGER'),
EMP_INFO(7788,
'SCOTT', 'ANALYST'), EMP_INFO(7876, 'ADAMS', 'CLERK'), EMP_INFO(7902, 'FORD',
'ANALYST'))

        30 SALES          CHICAGO
EMPS(EMP_INFO(7499, 'ALLEN', 'SALESMAN'), EMP_INFO(7521, 'WARD', 'SALESMAN'),
EMP_INFO(765
4, 'MARTIN', 'SALESMAN'), EMP_INFO(7698, 'BLAKE', 'MANAGER'), EMP_INFO(7844,
'TURNER', 'SA
LESMAN'), EMP_INFO(7900, 'JAMES', 'CLERK'))

        40 OPERATIONS     BOSTON
EMPS()
```

Another example is the instructor/courses object. I would never design a system like this. This is very simply a parent/child relationship and should be modeled as such.

Object Views

Where I would be more likely to model the above types is in an object view. If a developer needed a report, or a screen, that would display all departments and the employees in them, it would be very easy to provide them a view:

Now we create the DEPT_INFO type.

```
CREATE TYPE dept_info AS OBJECT (
   deptno NUMBER(2),
   dname VARCHAR2(14),
   loc VARCHAR2(13),
   emp_data emps );
 /
```

Notice the embedded EMPS type.

We create the view.

```
CREATE VIEW dept_view OF dept_info
  WITH OBJECT IDENTIFIER (deptno) AS
  SELECT deptno, dname, loc,
       CAST( MULTISET (
         SELECT empno, ename, job
           FROM emp
           WHERE emp.deptno = dept.deptno)
           AS emps)
           AS emp_info
   FROM dept
   /
```

The WITH OBJECT IDENTIFIER is defining the key or REF (OID) of the view. The syntax is required for an object view but I have never made use of it.

```
SQL> select * from dept_view;

    DEPTNO DNAME          LOC
---------- -------------- -------------
EMP_DATA(EMPNO, ENAME, JOB)
-----------------------------------------------------------------------------------
-------
        10 ACCOUNTING     NEW YORK
EMPS(EMP_INFO(7782, 'CLARK', 'MANAGER'), EMP_INFO(7839, 'KING', 'PRESIDENT'),
EMP_INFO(793
4, 'MILLER', 'CLERK'))

        20 RESEARCH       DALLAS
EMPS(EMP_INFO(7369, 'SMITH', 'CLERK'), EMP_INFO(7566, 'JONES', 'MANAGER'),
EMP_INFO(7788,
'SCOTT', 'ANALYST'), EMP_INFO(7876, 'ADAMS', 'CLERK'), EMP_INFO(7902, 'FORD',
'ANALYST'))

        30 SALES          CHICAGO
EMPS(EMP_INFO(7499, 'ALLEN', 'SALESMAN'), EMP_INFO(7521, 'WARD', 'SALESMAN'),
EMP_INFO(765
4, 'MARTIN', 'SALESMAN'), EMP_INFO(7698, 'BLAKE', 'MANAGER'), EMP_INFO(7844,
'TURNER', 'SA
LESMAN'), EMP_INFO(7900, 'JAMES', 'CLERK'))

        40 OPERATIONS     BOSTON
EMPS()
```

Same results as the object table but the data is stored relationally. Give your developers access to these types and this view and the report or screen almost writes itself. If they need to update this data, an instead of trigger is easily added.

```
CREATE OR REPLACE TRIGGER DEPT_VIEW_TRG
  INSTEAD OF DELETE OR INSERT OR UPDATE
  ON DEPT_VIEW
  FOR EACH ROW
    BEGIN
      dbms_output.put_line('Place update/insert/delete logic here');
    END dept_view_trg;
/
```

Besides providing data to clients, object views are also nice for serving up XML.

Use the Object View from above to create XML (see XML In the Database for xmlElement and XMLForest examples):

```
select xmlElement("Dept_Data",
         xmlForest(deptno, dname, emp_data)
       )
  from dept_view;
```

Run the query:

```
SQL> /

XMLELEMENT("DEPT_DATA",XMLFOREST(DEPTNO,DNAME,EMP_DATA))
-----------------------------------------------------------------------------------
<Dept_Data><DEPTNO>10</DEPTNO><DNAME>ACCOUNTING</DNAME><EMP_DATA><EMP_INFO><EMPN
<Dept_Data><DEPTNO>20</DEPTNO><DNAME>RESEARCH</DNAME><EMP_DATA><EMP_INFO><EMPNO>
<Dept_Data><DEPTNO>30</DEPTNO><DNAME>SALES</DNAME><EMP_DATA><EMP_INFO><EMPNO>749
<Dept_Data><DEPTNO>40</DEPTNO><DNAME>OPERATIONS</DNAME><EMP_DATA/></Dept_Data>
Let's look closer at the first row returned:
<Dept_Data>
  <DEPTNO>10</DEPTNO>
  <DNAME>ACCOUNTING</DNAME>
  <EMP_DATA>
    <EMP_INFO>
      <EMPNO>7782</EMPNO>
      <ENAME>CLARK</ENAME>
      <JOB>MANAGER</JOB>
    </EMP_INFO>
    <EMP_INFO>
      <EMPNO>7839</EMPNO>
      <ENAME>KING</ENAME>
      <JOB>PRESIDENT</JOB>
    </EMP_INFO>
    <EMP_INFO>
       <EMPNO>7934</EMPNO>
       <ENAME>MILLER</ENAME>
       <JOB>CLERK</JOB>
    </EMP_INFO>
  </EMP_DATA>
</Dept_Data>
```

Notice that Oracle understands both the object and the XML expectations and automatically nests the employee data.

And in the last line:

```
<Dept_Data>
  <DEPTNO>40</DEPTNO>
  <DNAME>OPERATIONS</DNAME>
  <EMP_DATA/>
</Dept_Data>
```

There are no employees in the operations department so Oracle gives us an empty tag.

It would be very easy to wrap our DEPT_INFO view in a new DEPT_INFO_XML view using the select above. We would then have a dual interface depending on what the requirements of the interface are.

An added bonus is that all of the SQL is in the database for tuning as well as hidden from the clients. Maintainable and tunable. Using objects. Who'd a thunk it?

And, honestly, that's the bulk of OO in Oracle. Between Parts 1,2,3 and 4, we've covered the majority of OO in Oracle. This entry turned out to be much longer than I intended, pagewise. If there is anything else on this topic that you would like covered, please post a comment here.

And just a quick note on my prototype: my local database has somewhat stabilized. The problem turned out to be a couple of AIX parameters that was giving the file buffers 80% of system memory. If I hadn't been running through a terabyte of data, that might have been ok.

French Quarter, New Orleans, LA – Christmas Day 2005

Chapter 8 - Collections

If builders built buildings the way programmers wrote programs, then the first woodpecker that came along would destroy civilization. - Weinberg's Second Law

Oracle Collections: A Definition in Plain English Part 1

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/005924.asp

Posted 9/27/2005

Today I want to talk about collections in Oracle. I'll explain what they are and give some examples on using them.

What is a collection?

A collection is nothing more than an array. The most basic collection is a single dimension array. Later on in this article, we will get into some examples of very complex arrays but we'll start with the simple stuff.

Associative Array

In Oracle 7, Oracle introduced an index by table, which was also called a PL/SQL table. This table was different from a SQL table in that it existed only in PL/SQL and you could NOT perform SQL against it.

The most basic example of an index by table:

```
TYPE a_number IS TABLE OF NUMBER INDEX BY BINARY_INTEGER.
```

This creates an array of unlimited size (up to your OS and DB version limitations) of NUMBER which is indexed by a BINARY_INTEGER datatype. The index is just the subscript and BINARY_INTEGER is just a numeric data type. An index by table does NOT have to be initialized.

Below is a small example showing how to create an array, create a variable from the array and then populate it with 5 values and display them in a loop:

```
DECLARE
  -- Declare the type
  TYPE a_char_data IS TABLE OF VARCHAR2(10) INDEX BY BINARY_INTEGER;

  --Declare a variable of that type
  v_char_data a_char_data;

BEGIN

  -- Populate some data
  v_char_data(1) := 'ABC';
  v_char_data(2) := 'DEF';
  v_char_data(3) := '1234567890';
  v_char_data(4) := 'X+Y=Z';
  v_char_data(5) := 'DE NADA';

  -- Loop thru the data and display via dbms_pitput
  FOR i IN v_char_data.FIRST..v_char_data.LAST LOOP
    dbms_output.put_line( v_char_data(i) );
  END LOOP;

END;
/
```

This example introduced two Collection Methods. If you've read my Sorcerer of OO series, you know that a method is just a function assigned to an object type. The FIRST method returns the first subscript in a collection, in this case a 1, and the LAST method returns the last subscript in a collection, in this case a 5. I'll talk about additional methods later in this article.

Records

Before I go any further with collections, I think it's important to understand another feature Oracle introduced in v7, records. You can think of a PL/SQL record just like you would the column layout in a table. A PL/SQL record is for use in PL/SQL programs.

```
DECLARE

  -- Declare the record type
  TYPE r_data IS RECORD (
    ssn VARCHAR2(9) NOT NULL := -1,
    name VARCHAR2(30),
    dob DATE );

  --Declare a variable of that type
  v_data r_data;

BEGIN

  -- Set some values
  v_data.ssn := '123456789';
  v_data.name := 'Lewis';
  v_data.dob := '01-JAN-1900';

END;
/
```

You can also declare a record using the structure of a table, i.e. SCOTT.EMP%ROWTYPE. That's a subject for a different discussion, however.

When you combine a record type with a collection type, you have a very powerful set of tools for building in memory data structures. With a minor change to the above examples, we can create a ragged, multi-dimensional array:

```
DECLARE
```

```
  -- Declare a basic table type type
  TYPE a_char_data IS TABLE OF VARCHAR2(10) INDEX BY BINARY_INTEGER;

  -- Declare a complex record type
  TYPE r_data IS RECORD (
    ssn VARCHAR2(9) NOT NULL := -1,
    name a_char_data,  -- Notice the table_type used here
    dob DATE );

  -- Declare a index by table using the complex record type
  TYPE a_multi IS TABLE OF r_data INDEX BY BINARY_INTEGER;

  -- Declare a variable using the complex array
  v_data a_multi;

BEGIN

  -- Set some values
  v_data(1).ssn := '123456789';
  v_data(1).dob := '01-JAN-1900';

  -- Notice the second subscript
  v_data(1).name(1) := 'Lewis';
  v_data(1).name(2) := 'Joe';

  dbms_output.put_line(v_data(1).ssn);

-- Loop through the v_data(1).name table

  FOR i IN v_data(1).name.FIRST..v_data(1).name.LAST LOOP

    dbms_output.put_line(v_data(1).name(i));

  END LOOP;

END;
/
```

Let me walk you through exactly what this example is doing:

- In the declare section:
  - First we declare our very basic array of VARCHAR2(10).
  - Next we declare a record type with an embedded index by table.
  - So now we have a ragged record type. A single record with a dimensional name column.
  - Then we declare a new table based on the complex record type. This creates an array of ragged records.
  - And finally, we create the complex variable.
- Executeable section:

- We populated the ssn and dob columns of the first record of the v_data variable.
- Next we populated the first and second rows of the name table in the first row of the v_data variable.
- Finally, we displayed the ssn of the first row of the v_data variable and then looped through the name table of the first row of the v_data variable.

Phew, again! If you are not quite getting this, stick with it. Cut and paste it into TOAD or PL/SQL Developer or something like that. It is confusing at first but it is so hugely powerful that it's worth the time.

Don't try this in v7 though. Oracle has continuously made improvements to both PL/SQL in general and to collections specifically. As a matter of a fact, it was a while before you could have anything but single dimension tables (you could not combine records and collections). And I'm not sure exactly which release added multi-dimensional tables but it was very recent.

Oracle has also added a new feature to collections that has been a long time coming. Associative arrays! Wait a minute. Isn't that just index by tables with a new name? Sort of. What they added was *index by varchar2*. So now you can have something like this:

```
DECLARE
  -- Declare the type using varchar2
  TYPE a_depts IS TABLE OF NUMBER INDEX BY VARCHAR2(10);

  --Declare a variable of that type
  v_depts a_depts;

  -- Declare a counter
  i VARCHAR2(10);

BEGIN

  -- Loop thru scott.emp
  FOR c_depts IN (
        SELECT deptno, ename
          FROM SCOTT.emp ) LOOP

    -- Load up the array with values from scott.emp
    v_depts(c_depts.ename) := c_depts.deptno;

  END LOOP;

  -- Initialize our CHAR counter, i.e. ename
  i := v_depts.FIRST;

  -- Loop thru the data
  LOOP
    -- Exit when the next value is null, i.e. does not exist
    EXIT WHEN NOT v_depts.EXISTS(i);
      dbms_output.put_line( i || ' is in dept: ' || to_char(v_depts(i)) );

      -- Increment the counter
      i := v_depts.next(i);
  END LOOP;

END;
/
```

I introduced two new collection methods here. EXISTS and NEXT. EXISTS accepts a subscript and returns TRUE if the element of the array at that subscript exists. NEXT returns the subscript of the next elements. PRIOR was not used here but it would return the subscript of the previous element in the array. By using PRIOR and NEXT, you can loop through arrays indexed by non-numeric data as well as loop through sparse arrays. I'll cover sparse arrays in one of the next entries on this topic.

I think this is probably enough to think about for today. I will be following this entry up with several more parts covering varrays, nested tables, bulk collect, forall, and several other very useful topics.

As usual, if you have any questions about this material or any suggestions about how I can improve this entry or my blog in general, please post a comment.

Oracle Collections: A Definition in Plain English Part 2

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006003.asp

Posted 10/4/2005

In part 1 of this series I covered Associative Arrays and Record types. Today I will be covering Varrays and Nested Tables.

Nested Table

So, what is the difference between an Associative Table and Nested Table? A nested table looks a lot like an associative table. The main difference in the declaration is the missing INDEX BY keywords.

If we go back to our first example in part 1, we can modify it easily to use a nested table instead:

```
DECLARE
  TYPE a_char_data IS TABLE OF VARCHAR2(10);

  v_char_data a_char_data := a_char_data();

BEGIN
  v_char_data.EXTEND;
  v_char_data(1) := 'ABC';
  v_char_data.EXTEND;
  v_char_data(2) := 'DEF';
  v_char_data.EXTEND;
  v_char_data(3) := '1234567890';
  v_char_data.EXTEND;
  v_char_data(4) := 'X+Y=Z';
  v_char_data.EXTEND;
  v_char_data(5) := 'DE NADA';

  FOR i IN v_char_data.FIRST..v_char_data.LAST LOOP
    dbms_output.put_line(v_char_data(i));
  END LOOP;
```

```
END;
/
```

There are three changes to note here.

- The TYPE declaration does not include the INDEX BY keyword.
- Notice the variable declaration for v_char_data now includes a call to a Nested Table CONSTRUCTOR. See the first part of my Sorcerer of OO for a description of constructors.
- And finally, I've introduced the EXTEND method. EXTEND adds a new, empty element to a nested table. You must call EXTEND before referencing an element or you will get an error.

VARRAY

So, what is the difference between a Nested Table and a Varray? Well, here is that same example using a Varray.

```
DECLARE
  TYPE a_char_data IS VARRAY(5) OF VARCHAR2(10);

  v_char_data a_char_data := a_char_data();

BEGIN
  v_char_data.EXTEND;
  v_char_data(1) := 'ABC';
  v_char_data.EXTEND;
  v_char_data(2) := 'DEF';
  v_char_data.EXTEND;
  v_char_data(3) := '1234567890';
  v_char_data.EXTEND;
  v_char_data(4) := 'X+Y=Z';
  v_char_data.EXTEND;
  v_char_data(5) := 'DE NADA';

  FOR i IN v_char_data.FIRST..v_char_data.LAST LOOP
    dbms_output.put_line(v_char_data(i));
  END LOOP;

END;
/
```

Not a lot of differences there. To declare a Varray, you just replace the TABLE OF keywords with VARRAY(size limit). In this case, I made it a maximum length of 5. What would happen if I made it 4 and then tried to add 5 elements? You get an error very much like you would if you didn't use Extend. Bascially, the element does not exist.

Varrays and Nested Tables in SQL

Associative tables cannot be used in SQL. Both Varrays and Nested tables can. There are two ways you can use collections in SQL.

Arrays in Tables

You can declare a column in a table as either like this example:

```
CREATE OR REPLACE TYPE addresses_v IS VARRAY(3) OF VARCHAR2(30);
/

CREATE OR REPLACE TYPE addresses_n IS TABLE OF VARCHAR2(30);
/

CREATE TABLE person (
  name VARCHAR2(30),
  dob DATE,
  address_v addresses_v,
  address_n addresses_n )
  NESTED TABLE address_n STORE AS address_nested
/

SQL> INSERT INTO person (name, dob, address_v, address_n)
  2    VALUES ('Lewis', to_date('01-jan-1960', 'dd-mon-yyyy'),
  3          addresses_v('Line 1', 'Line 2', 'Line 3' ),
  4          addresses_n('Line 1', 'Line 2', 'Line 3' )
  5          )
  6  /

1 row created.

SQL>
SQL> SELECT * FROM person
  2  /

NAME                           DOB
------------------------------ ---------
ADDRESS_V
------------------------------------------------------------------
ADDRESS_N
------------------------------------------------------------------
Lewis                          01-JAN-60
ADDRESSES_V('Line 1', 'Line 2', 'Line 3')
ADDRESSES_N('Line 1', 'Line 2', 'Line 3')


SQL>
SQL> UPDATE TABLE(select address_n from person)
  2    SET column_value = 'Line #2'
  3    WHERE column_value = 'Line 2'
  4  /

1 row updated.

SQL>
SQL> SELECT * FROM person
  2  /

NAME                           DOB
```

```
------------------------------ ---------
ADDRESS_V
-----------------------------------------------------------------
ADDRESS_N
-----------------------------------------------------------------
Lewis                          01-JAN-60
ADDRESSES_V('Line 1', 'Line 2', 'Line 3')
ADDRESSES_N('Line 1', 'Line #2', 'Line 3')


SQL>
SQL> UPDATE person
  2    SET address_v = addresses_v('Line 3', 'Line 2', 'Line 1' )
  3  /

1 row updated.

SQL>
SQL> SELECT * FROM person
  2  /

NAME                           DOB
------------------------------ ---------
ADDRESS_V
-----------------------------------------------------------------
ADDRESS_N
-----------------------------------------------------------------
Lewis                          01-JAN-60
ADDRESSES_V('Line 3', 'Line 2', 'Line 1')
ADDRESSES_N('Line 1', 'Line #2', 'Line 3')


SQL>
```

Tables from Arrays

You can also use a collection in your SQL like this:

```
CREATE TABLE data_and_stuff (
  name VARCHAR2(30),
  dob DATE )
/

INSERT INTO data_and_stuff values ('LEWIS', sysdate - 1 );
INSERT INTO data_and_stuff values ('DAVID', sysdate - 2 );
INSERT INTO data_and_stuff values ('EILEEN', sysdate - 3 );
INSERT INTO data_and_stuff values ('ERICK', sysdate - 4 );

CREATE OR REPLACE TYPE a_data IS TABLE OF VARCHAR2(30);
/

SQL> set serveroutput on size 200000
SQL> DECLARE
  2
  3    v_data a_data := a_data( 'TIM', 'DAVID', 'ERICK', 'LEWIS' );
```

```
BEGIN

  FOR c1 IN (
    SELECT * FROM TABLE(CAST( v_data AS a_data)) ) LOOP

    DBMS_OUTPUT.PUT_LINE('Looking for values: ' || c1.column_value);
  END LOOP;

  DBMS_OUTPUT.PUT_LINE('Begin search: ' );

  FOR c1 IN (
      SELECT *
        FROM data_and_stuff
        WHERE name IN (
          SELECT * FROM TABLE(CAST(v_data AS a_data)) ) ) LOOP

    DBMS_OUTPUT.PUT_LINE(c1.name || ' was born on ' ||
            to_char( c1.dob, 'DD-MON-YYYY' ) );

  END LOOP;

END;
/
```

```
Looking for values: TIM
Looking for values: DAVID
Looking for values: ERICK
Looking for values: LEWIS
Begin search:
DAVID was born on 30-SEP-2005
ERICK was born on 28-SEP-2005
LEWIS was born on 01-OCT-2005

PL/SQL procedure successfully completed.
```

Using the above logic, you can allow a parameter to a function as a nested table and use it directly in your SQL. This is a very powerful concept.

If you would like to see how to use Collections in views and how to use CAST and MULTISET, be sure to read the Sorcerer of OO, Part 4.

When to use which

Now I guess we should talk about when to use a particular type of collection. In PL/SQL, for the most part, it's a matter of preference. If you know you have a maximum, use VARRAY. The choice between Nested Table and Associative Table is not as clear cut. In general, I use Nested Table instead of Associative Tables unless I am indexing the array with a VARCHAR2.

In SQL, it's a different subject. Of course, you cannot use an Associative table at all in SQL so your choices are down to Nested Table or Varray. I rarely, if ever, use nested tables or varrays in a relational table. I prefer to create a normal table and use foreign keys.

And that's it for today. In the next entry in this series, I will cover sparse arrays, bulk collect and forall.

Oracle Collections: A Definition in Plain English Part 3

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006098.asp

Posted 10/11/2005

In part 1 of this series I covered Associative Arrays and Record types. In Part 2, I covered Varrays and Nested Tables. Today I will be covering Sparse Arrays, Bulk Collect and the FORALL command.

Sparse Arrays

A sparse array is just an array that has empty elements between other elements. For example, if you run the code below, you will get an error of NO DATA FOUND.

```
DECLARE
  TYPE a_char_data IS TABLE OF VARCHAR2(10);

  v_char_data a_char_data := a_char_data();

BEGIN
  v_char_data.EXTEND;
  v_char_data(1) := 'ABC';
  v_char_data.EXTEND;
  v_char_data(2) := 'DEF';
  v_char_data.EXTEND;
  v_char_data(3) := '1234567890';
  v_char_data.EXTEND;
  v_char_data(4) := 'X+Y=Z';
  v_char_data.EXTEND;
  v_char_data(5) := 'DE NADA';

  v_char_data.DELETE(3);

  FOR i IN v_char_data.FIRST..v_char_data.LAST LOOP
    dbms_output.put_line(v_char_data(i));
  END LOOP;

END;
/
```

You get a NO DATA FOUND because by using the DELETE method, you now have a sparse array. Your index jumps from 2 to 4. Nested Tables and Index By Tables can be sparse. A VARRAY cannot.

You can also use the DELETE method to remove a range of elements, i.e. v_char_data.DELETE(2,4) will delete elements 2, 3 and 4.

So, if you can have sparse arrays, how will you loop through it with getting an exception? I mentioned the solution briefly in part 1 of this series. You can use the PRIOR and NEXT methods to loop. Here is the same example as above, but it will now execute successfully:

```
DECLARE
  TYPE a_char_data IS TABLE OF VARCHAR2(10);
  v_char_data a_char_data := a_char_data();

  v_cnt NUMBER := 0;
BEGIN
  v_char_data.EXTEND;
  v_char_data(1) := 'ABC';
  v_char_data.EXTEND;
  v_char_data(2) := 'DEF';
  v_char_data.EXTEND;
  v_char_data(3) := '1234567890';
  v_char_data.EXTEND;
  v_char_data(4) := 'X+Y=Z';
  v_char_data.EXTEND;
  v_char_data(5) := 'DE NADA';

  v_char_data.DELETE(3);

  LOOP
    EXIT WHEN v_cnt >= v_char_data.LAST;
    v_cnt := v_char_data.NEXT(v_cnt);
    dbms_output.put_line(v_char_data(v_cnt));
  END LOOP;

END;
/
```

You can loop backwards through the array by using the PRIOR method.

Bulk Collect

A powerful feature of PL/SQL and collections is the ability to SELECT INTO a collection. Instead of slowly looping through your data row by row, you can select into an array and then process that array.

The steps to do this are simple. Simply declare an array of records, select into it and then process the records. Here is an example:

```
DECLARE

  TYPE a_recs IS TABLE OF emp%ROWTYPE;

  v_recs a_recs;

BEGIN

  SELECT *
    BULK COLLECT INTO v_recs
    FROM emp;

  FOR i IN v_recs.FIRST..v_recs.LAST LOOP
    DBMS_OUTPUT.PUT_LINE( v_recs(i).ename ||
                          ' belongs to deptno: ' ||
                          to_char(v_recs(i).deptno ) );
  END LOOP;
```

```
END;
/
```

And that really is the guts of the BULK COLLECT. I have had a lot of people express some confusion over how to get this working.

Here is the same code using a cursor for fetching. It's a little bit more complex but is more of a real world example.

```
DECLARE

  TYPE a_recs IS TABLE OF emp%ROWTYPE;

  v_recs a_recs;

  CURSOR c1 IS
    SELECT *
    FROM emp;

BEGIN

  OPEN c1;

  LOOP
    EXIT WHEN c1%NOTFOUND;

    FETCH c1 BULK COLLECT INTO v_recs LIMIT 10;

    FOR i IN v_recs.FIRST..v_recs.LAST LOOP
      DBMS_OUTPUT.PUT_LINE( v_recs(i).ename ||
                            ' belongs to deptno: ' ||
                            to_char(v_recs(i).deptno ) );
    END LOOP;

  END LOOP;

END;
```

Notice the addition of the LIMIT keyword. That will keep Oracle from selecting more than that number of records at a time. Why would you want to limit it? If you are processing 1 million rows, you want to work on chunks of data at a time. Remember that more is not always better. Start with 100 and test. Bump it up and test. At some point, you will hit the point of diminishing returns. Also, remember that you will probably be running in a multi-user system. Don't try to suck up all of the system memory.

I guess another question some people have about BULK COLLECT is why add the extra coding if a CURSOR FOR loop would work just as well. Each time Oracle has to switch between SQL and PL/SQL, called a context switch, there is overhead. If you are working with millions of rows, that overhead adds up to real numbers. By bulk collecting 100 or 1000 rows at a time, you reduce the overhead by that much.

FORALL

The FORALL keyword allows a program to perform Insert, Update and Delete operations against a collection. This also reduces the overhead of switching between SQL and PL/SQL. I'll use our previous example to copy the data from the SCOTT.EMP table into a backup table.

First, we create the back up table:

```
CREATE TABLE emp_bkup
AS SELECT * from emp
  WHERE rownum != 0;
```

And then run our modified code:

```
DECLARE

  TYPE a_recs IS TABLE OF emp%ROWTYPE;

  v_recs a_recs;

  CURSOR c1 IS
    SELECT *
    FROM emp;

BEGIN

  OPEN c1;

  LOOP
    EXIT WHEN c1%NOTFOUND;

    FETCH c1 BULK COLLECT INTO v_recs LIMIT 10;

    v_recs.DELETE(6);

    BEGIN
      FORALL i IN v_recs.FIRST..v_recs.LAST
        SAVE EXCEPTIONS
        INSERT INTO emp_bkup
          VALUES v_recs(i);

    EXCEPTION
      WHEN OTHERS THEN  -- Bad Practice Alert, Don't use others
        FOR i IN 1..SQL%BULK_EXCEPTIONS.COUNT LOOP
          DBMS_OUTPUT.PUT_LINE('Found an error at ' ||
                SQL%BULK_EXCEPTIONS(i).ERROR_INDEX ||
                ' Error Msg: ' ||
                SQLERRM(-SQL%BULK_EXCEPTIONS(i).ERROR_CODE) );
        END LOOP;
    END;

  END LOOP;

END;
/
```

This is mostly the same as the previous example. Two new pieces were added:

```
      FORALL i IN v_recs.FIRST..v_recs.LAST
        SAVE EXCEPTIONS
        INSERT INTO emp_bkup
          VALUES v_recs(i);
```

This code runs the INSERT command for all records in the nested table and only has a single context switch per set of data (each loop to load the array is a switch).

If you notice above, I added a

```
    v_recs.DELETE(6);
```

I know that there are 14 records in the SCOTT.EMP table. I wanted to show a single exception in the exception section. What happens when we use FIRST and LAST with an index to loop through a sparse array? You're right, an exception.

So I put the FORALL in it's own exception block and I handle the exception I created as follows:

```
      WHEN OTHERS THEN  -- Bad Practice Alert, Don't use others
        FOR i IN 1..SQL%BULK_EXCEPTIONS.COUNT LOOP
          DBMS_OUTPUT.PUT_LINE('Found an error at ' ||
                SQL%BULK_EXCEPTIONS(i).ERROR_INDEX ||
                ' Error Msg: ' ||
                SQLERRM(-SQL%BULK_EXCEPTIONS(i).ERROR_CODE) );
        END LOOP;
```

In the exception block, I loop through a nested table called BULK_EXCEPTIONS. This table is IMPLICITLY created when I used the keywords SAVE EXCEPTIONS. If I had not included SAVE EXCEPTIONS, the error would have been raised and the code would have failed.

To see how this would work without capturing the exceptions, you can run this code:

```
DECLARE
  TYPE a_recs IS TABLE OF emp%ROWTYPE;

  v_recs a_recs;

  CURSOR c1 IS
    SELECT *
    FROM emp;

BEGIN

  OPEN c1;

  LOOP
    EXIT WHEN c1%NOTFOUND;

    FETCH c1 BULK COLLECT INTO v_recs LIMIT 10;

    v_recs.DELETE(6);
    FORALL i IN v_recs.FIRST..v_recs.LAST
      INSERT INTO emp_bkup
        VALUES v_recs(i);

  END LOOP;

END;
/
```

Pretty intense stuff here. Very powerful. Coming soon will be an article on the advanced usage of Bulk Collect and Forall. These two features are very powerful, especially in a warehousing environment.

10gR2 has added a few more really neat features for getting data into a database. I'll cover those after I've had a chance to play with them some more.

Downtown New Orleans, La - 11am, Friday, December 23rd, 2005

CHAPTER 9 - SECURITY

Too many people are thinking of security instead of opportunity. They seem more afraid of life than death.
- James F. Byrnes

Oracle 10g Security and Audit

Original Post: http://security.ittoolbox.com/white-papers/oracle-10g-security-and-audit-5063

Posted 10/07/2005

Introduction

What exactly do I mean by security? There are several aspects of security in an enterprise work place: locked doors, guards, cameras, etc. There are just as many aspects when protecting enterprise data. Oracle provides security in four ways: authentication, authorization, data access and audit.

Authentication

Authentication is a verification that you are who you say you are. It's the equivalent of showing a guard your ID. By default, Oracle implements authentication through a user ID and password; this is called database authentication.

Another method of authentication is OS authentication. Using this method, a user logs into the OS and Oracle will accept that authentication as gospel. Sometimes you will hear OS authentication referred to as OPS$ (pronounced "ops dollar") accounts. When you use OS authentication to login into SQL*Plus, the command line looks something like "sqlplus / ". When you use a "/", Oracle looks for an ID that matches your OS login ID and prepends an OPS$ to it. So if your OS account is lcunning, your Oracle account would look like OPS$lcunning. The OPS$ is just a default and can be changed.

OPS$ accounts provide an additional benefit. OS scripts can log into the database without hard coding user names and passwords. This enhances overall database security.

In my experience, database and OS authentication are the most common authentication methods. However, I have run into several other methods. You can tie authentication to third-party providers like Kerberos or DCE (called network authentication) or provide it from the middle tier (called multitier authentication).

Network authentication is more expensive as you must be running Oracle Enterprise server and license an additional security pack.

There are several security issues relating to multitier security due to its distributed nature. The client must authenticate to the middle tier and the middle tier must authenticate to the database. In addition, because multiple users are sharing a connection, the database must be able to distinguish the application from the user and one user from another. This white paper does not address multitier security.

Authorization

Authorization is giving access to certain objects, processes or resources. The equivalent in our enterprise analogy would be a key card allowing access to a specific room or having the pass code to the VIP restroom (I want one of those. I bet all the seats have felt covers and it smells like roses and lavender in there). The thought here being that management has granted you access to these rooms or resources.

Authorization is implemented using GRANTS on objects to ROLES or USERS. Let's say your ID is JohnD. Your friendly neighborhood DBA can grant you SELECT access on the EMP table owned by SCOTT. Or not. You never can tell about those DBA people. But if the DBA will grant you that access, from that point on you can select from SCOTT.EMP. Alternatively, the DBA can create a ROLE called EMP_DEPT and grant SELECT on EMP and DEPT to EMP_DEPT. If he then granted the emp_dept role to your ID, you would be able to select from both the EMP and DEPT tables.

So, the short story here is that authorization is getting access to specific objects. Oracle implements authorization via users, roles and permissions. Combined with Oracle's offerings for data access, this provides for a very secure database.

Data Access

Data Access is the ability to access specific data values associated with an object. Data access security can be partitioned across records and/or across columns. Keeping with my previous analogy, if the guard let you into the building and your key let you into the VIP restroom, your access will only allow you to use the unlocked stalls. Sometimes it feels like data access is security that allows you to use the stall but not the toilet paper (TP) dispenser. Ok, that really is a bad analogy but it's the best I can think of at the moment. But, I think that gets the point across.

Using stored procedures is a common methodology to secure data. If all access to the data is through stored procedures, it's easy to return only certain columns or records to the calling program. However, if you have a requirement for ad hoc access to data, this is not realistic. I do recommend that all programmatic data access be implemented using procedures. In addition to securing the data, it protects the application by hiding the structure of the data. However, that's a completely different article.

The common method to secure data in specific columns is through secure views. Instead of granting access to a table, you can create a view and grant access to that. The view can contain all or a subset of the columns in a table. You would want to use the WITH CHECK OPTION constraint if the view allows updates or deletes and you've restricted access to certain rows.

In older versions of Oracle, record level access was also implemented using secure views. In more recent versions though, Oracle has introduced Virtual Private Databases (VPD) and, even more recently, Fine Grained Access Control (FGAC). This is implemented using a package called DBMS_RLS (RLS = Row Level Security) and security contexts. The package basically let's you put a where clause on any combinations of DML and SELECT so that when a user queries the database, the where clause is added behind the scenes. The context could be a department ID or day of week or any other criteria you would like to use to secure the data. The actual building of the where clause is done with PL/SQL via a secure package.

Audit

Audit is keeping track of who did what and when. Security cameras and tracking electronic key cards through a building completes my analogy. Or to really complete it, a sensor showing how much TP was used. Hey, when I find a bad analogy, I stick to it!

Obviously, an audit trail is used for after the fact research and will not protect your data. It will let you know what someone is doing with or to your data.

In older versions of Oracle (and even now for some reason, some DBAs want to use the old methods), auditing was implemented using triggers. For DML, a trigger would fire and copy data to an audit table or audit log. You can still implement audit that way but there are downsides to it. The most basic is that a trigger can easily be disabled and without correct permissions in place, audit tables can be modified. Triggers also do not fire on SELECT so you are not able to track what data is being selected by whom.

Oracle has provided built-in audit capabilities for many years and in the most recent versions, this auditing is very sophisticated. The Fine Grained Auditing (FGA) that you can use now can provide audits for all DML against a database, a schema or specific objects. FGA can also audit who was selecting data and not just who updated data. Oracle can also track who changed database objects (DDL, like alter table or compile procedure), what they did and when they did it.

Trigger based auditing requires extensive coding and testing. Depending on the detail you need, Oracle Audit can be implemented with little to no coding and testing is easier because of that.

FGA is implemented using a package called DBMS_FGA. Using DBMS_FGA you define rule based events that can store off data or even email or page someone that will identify what rule was broken and who broke it. The rules can be table driven or even driven by the client, tier or OS environments. FGA combined with LogMiner can actually reproduce the exact records returned by a user's query.

The Technical Details of Security

We need to create an account for Jane Doe. Ok, following our naming standards (and we do have naming standards, right?), you will be JDoe in the application.

```
CREATE USER jdoe IDENTIFIED BY "Strong1!Password";
  GRANT connect TO jdoe;
```

If we wanted to create this user to use OS authentication, we would do:

```
CREATE USER jdoe IDENTIFIED EXTERNALLY;
GRANT connect TO jdoe;
```

"CREATE USER", oddly enough, creates a user. By default, Oracle usernames and passwords are not case sensitive. I'll talk more about fixing that in part three.

"IDENTIFIED" describes the password. Using the keyword "BY" tells oracle that you will define the users password, "Strong1!Password". "EXTERNALLY" lets oracle know to create the user with out a local password and to accept the validity of the user if the OS authenticates the person.

What is a grant? A grant allows something. "GRANT CONNECT" is a special permission that allows the user to connect to the database. At this point, jdoe can do nothing but login. If we said "GRANT

SELECT ON emp", assuming that a table or view named emp existed, jdoe would now be able to select records from that object. REVOKE is the opposite of GRANT. "REVOKE connect FROM jdoe" is an important command that we will talk more about next entry.

So now we have a USER named jdoe. What's the difference between a user and a schema? I have always thought that it was kind of confusing that people seem to use user and schema interchangeably. A user is not the same as a schema. A user is an ID that allows access to a database and a schema is all of the objects created using that ID.

What is a role? A role is similar to a user. The biggest differences are that a role cannot login to the database and a role cannot own objects in the database. A role is just an object that can contain permissions. You can GRANT the role access to a table or a view or give the role the ability to execute a stored procedure. Nothing can happen with that role until you grant the role to a user. Once you grant the role to a user, that user inherits all of the permissions assigned to the role.

So how do we secure our data? Let's assume that we have a table, EMP, and that table contains sensitive information. Here is the definition of EMP:

```
SQL> desc EMP
Name                 Null?        Type
-------------------- ------------ -------------
EMPNO                NOT NULL     NUMBER(4)
ENAME                             VARCHAR2(10)
JOB                               VARCHAR2(9)
MGR                               NUMBER(4)
HIREDATE                          DATE
SAL                               NUMBER(7,2)
COMM                              NUMBER(7,2)
DEPTNO                            NUMBER(2)
```

Jdoe should not be able to see the SAL or COMM columns. This is easily done.

In case we have already granted anything on EMP to jdoe, we'll first revoke access to it.

```
REVOKE all ON EMP FROM jdoe;
```

jdoe no longer has access to EMP.

Now let's create a view that has the columns jdoe can see and grant her access:

```
CREATE OR REPLACE VIEW emp_no_sal AS
  SELECT EMPNO, ENAME, JOB, MGR, HIREDATE, DEPTNO
    FROM EMP;

GRANT select ON emp_no_sal TO jdoe;
```

That is all it takes to secure particular columns from prying eyes. This is so easy to do and in my opinion so under utilized that it hurts. I've seen a table split off into different tables or sql access being forbidden all because some basic design was not done.

The first statement creates a view. A view is just a stored query. When jdoe selects from the view, Oracle runs the query stored in the view and returns those columns. The GRANT is giving jdoe select access to the view.

Ok, that hides a column but jdoe shouldn't be able to see any manager's records at all. How would we do that? Well, if a mgr is defined as a record with an empno stored in the mgr field, we could re-write the query as:

```
CREATE OR REPLACE VIEW emp_no_sal AS
  SELECT EMPNO, ENAME, JOB, MGR, HIREDATE, DEPTNO
    FROM EMP a
    WHERE empno NOT IN (
          SELECT nvl(mgr,-1)
            FROM emp b );

GRANT select ON emp_no_sal TO jdoe;
```

jdoe can now only select from this view, not the underlying table, and she can only see the records not defined as a manager.

Just a quick note here, with Virtual Private Databases (VPDs), Oracle allows column masking to hide columns. When you select the row, Oracle will only display NULL for the secure columns. If you're securing at the row level and column level, it's probably easier to just implement VPDs and not the secure views. However, when people can see a column and it looks like it's null they're may be confusion. I spent quite a bit of time one day try to figure out where my data went before I remembered column masking. Doi!

So how can we secure our entire database for all DML as opposed to writing views for a few tables? Now we're looking at a solution like VPD.

What is a VPD? A VPD is just asking Oracle to put a where clause on DML against an object with a security policy on it. A security policy is defined with DBMS_RLS. For example, we could restrict access to the EMP table so that no one can see any records in the EMP table with a DEPT of 20.

A security policy is normally defined in a context. That is really what gives it the power. A context is just a piece of data that says how the where clause should be built. A context is set by a secure procedure.

Here is how we define our, very simplistic, example:

```
CREATE OR REPLACE PACKAGE scott.scott_dept_context AS
  PROCEDURE set_dept;
END;
/

CREATE OR REPLACE PACKAGE BODY scott.scott_dept_context AS
  PROCEDURE set_dept IS
  BEGIN
    DBMS_SESSION.SET_CONTEXT('which_depts', 'dept', 20);
  END set_dept;
END;
/
```

This creates a package that can be called at login time to set the context. In this case, we are populating the context which_depts and setting a variable called dept with the value of 20. The set context could be, and normally is, completely dynamic. Instead of a hard-coded 20, it would normally be a select against some kind of application table.

```
CREATE CONTEXT which_depts USING scott_dept_context;
```

This creates our context variable. This context is available to everyone in the database but the values inside it are set by the user's session.

The scott_dept_context defines the procedure that will set the context. We defined that package above.

```
CREATE OR REPLACE PACKAGE dept_pkg AS
  FUNCTION secure(p_owner VARCHAR2, p_object VARCHAR2)
      RETURN VARCHAR2;
END;
/

CREATE OR REPLACE PACKAGE BODY dept_pkg AS

  FUNCTION secure(p_owner VARCHAR2, p_object VARCHAR2)
           RETURN VARCHAR2 IS

    v_where VARCHAR2 (2000);

  BEGIN

    v_where := 'dept_no = SYS_CONTEXT(
                  ''scott_dept_conext'', ''dept_no'')';

    RETURN v_where;

  END;
END;
/
```

This package will set the where clause when the policy is invoked. Notice the v_where value is enclosed in quotes. This is because this clause will be dynamically added to the SQL that is being run.

Now we create the security policy on the object to be secured, in this case our EMP table.

```
BEGIN
  DBMS_RLS.ADD_POLICY (
         object_schema => 'scott',
         object_name => 'emp',
         policy_name => 'dept_pl',
         policy_function => 'dept_pkg.secure',
         statament_types => 'select');
END;
/
```

The parameters used are:

- object_schema - user that owns the objects
- object_name - What object
- policy_name - What is the name of this policy

- policy_function - What secure procedure will build the where clause
- statament_types - What kind of DML does this apply to

So what we said with that DBMS_RLS command is to create a policy that will secure select statements on the emp table in the scott schema and it will use the package dept_pkg.secure to build the where clause.

And now, a select against the EMP table will return all rows except those with a dept of 20.

Seems like a lot of work but imagine if you had a dept ID in every table in your schema (or at least all of the driving tables). When a user logged in you could use a login trigger to set the context to their dept ID. Using a policy on each table and a single package, you could secure the entire schema so that users could only see the data in their own dept ID. This would be enforced no matter what application was accessing the database. All screens, reports, etc would automatically get the same security.

Audit

Regardless of any regulations you are required to meet, you can implement a security and audit scheme that will cover you for most, if not all, eventualities. Having good security and auditability should be desirable regardless of regulations. It protects your business.

So what do I mean by Audit? Well, auditing is knowing what's going on in your environment. Future research requirements, accountability, error correction, etc are all covered by audit.

So how does Oracle implement audit? There are several ways you can do it: Standard Audit, Fine Grained Audit and Trigger Based Audit.

Standard Audit

Standard auditing is the basic audit capabilities of Oracle. Without going too far into it, Oracle will allow you to audit the actions of certain users, certain objects or certain permissions in the database. It's highly configurable and can write to a table in the database or to a file in the operating system. This method is very performant as each statement is audited in the parse phase of execution. The downside to this efficient form of auditing is that the data you can capture is limited to predefined columns.

Fine Grained Audit (FGA)

FGA is content based auditing. For example, you can audit a select statement on the EMP table where someone queries all records with a SAL > 10000. Earlier, I described fine-grained access control (FGAC), which was defined by DBMS_RLS. FGA is much like that. It is defined by a package called DBMS_FGA. You add a policy, for select, update, insert, or delete and define which columns and conditions to monitor. In the policy, you also identify the packaged procedure to execute when the policy condition is met. The entire process looks a lot like the security policies described above.

Trigger Based

This is the good old boy of auditing. Why use triggers? Some DBAs will not turn on auditing because of FUD (Fear, Uncertainty and Doubt). Or, maybe you need some very specific information not easily captured otherwise. Anyway, the way to implement this is to create an audit table, usually looking just like an existing table and put an "insert and update and delete" trigger on it to always copy the existing record to the audit table. You can save some i/o by just using an "update and delete" trigger. You already know what the inserted record looks like since it's in the master table.

So, when do you use which form of audit? That's actually a fairly easy issue to decide. Standard audit is going to be the most performant. It can catch general data fishing expeditions (credit card info for example) as well as people trying to execute permissions beyond their grants or against objects they shouldn't be. For regular proof of access type auditing, i.e. SOX or SEI's CMM, standard auditing should be the preferred method.

If you need to track access by content, use FGA. If you want to monitor who is looking at executive salaries, use FGA. If the requirement can be spelled out in a SQL where clause, use FGA. If standard audit can answer the requirement, do not use FGA.

And finally, in 10g, I see no reason to use trigger-based audits. Between the functionality provided by standard audit and FGA, triggers provide nothing new. They are also the least performant and the most expensive to maintain.

The Technical Details of Audit

This section is a brief, technical discussion about the implementation of audit in Oracle 10g. I'll quickly show how to turn on standard audit and how to see some of the audit taking place. I'll follow that with an implementation of FGA and finally a trigger based audit.

One aspect of auditing is to audit your administrators. This is a log to track what your DBA type folks might be doing in the database. You have to allow your DBAs full access to your database. Trying to hide things from them is nearly impossible and will hurt you when they try to debug system issues. The best thing to do is audit them.

SYS (meaning the sys database user and SYSDBA) auditing will record every SQL run by anyone connected as a system DBA user.

To turn on audit in the database you use an initialization parameter: AUDIT_SYS_OPERATIONS. Whenever this parameter is changed, the database must be stopped and restarted. To ensure that the parameter is not changed, you should have a process in place to verify the parameter on database restart.

SYS audit records are always written to the OS. DBAs can maintain the DB audit tables (as they can get very large, very fast) but you can secure the OS audit log from DBA access. In Windows, the audit log is actually the system event viewer. This is a graphical event log. It's a nice little feature actually.

Standard Audit

Ok. Now on to standard audit. Standard audit is the ability to audit a users object access and statement use. You can also audit on system privileges but I have not found that to be very useful. In general, for basic audit and compliance, you will use object audits.

An object audit is auditing everyone's access to a particular table or view. In my example, I will use the EMP table in the SCOTT schema.

Before we can audit records, we must turn on the audit trail. You turn on the audit trail by setting an initialization parameter, AUDIT_TRAIL. This is another parameter that requires the database to be restarted. You can choose to audit to a table in the database or to an OS file. I chose database:

```
SQL> alter system set audit_trail = DB scope = spfile;
```

Once you've set the parameter, you can issue audit statements.

```
SYS-SQL> audit insert, delete on scott.emp by access;
```

This statement will audit any insert or delete statements against the EMP table, which is owned by scott.

If I insert a record as SCOTT:

```
SCOTT-SQL> insert into emp (empno) values (-1)
/
```

I can query the audit trail:

```
SYS-SQL> /
NAME       ACTION     OBJNAME
---------- ---------- ----------
SCOTT      INSERT     EMP
```

I update the table as SCOTT:

```
SCOTT-SQL> update emp set ename = 'lewis'
        2 where empno = -1;

1 row updated.
```

There is no change to the audit as I am not auditing update:

```
SYS-SQL> /
NAME       ACTION     OBJNAME
---------- ---------- ----------
SCOTT      INSERT     EMP
```

Now, try a delete:

```
SCOTT-SQL> delete from emp where empno = -1;

1 row deleted.
```

Rollback the delete.

```
SCOTT-SQL> rollback;

Rollback complete.
```

And even though we rolled back the change, we still have the delete entry:

```
SYS-SQL> /
```

```
NAME       ACTION     OBJNAME
---------- ---------- ----------
SCOTT      INSERT     EMP
SCOTT      DELETE     EMP
```

If you're wondering what query I was running, it is:

```
select substr(userid,1,10) name,
       substr(name,1,10) action,
       substr(obj$name,1,10) objName
  from aud$, audit_actions
  where action = action#
/
```

Instead of this query, you can use views provided by oracle. There are several and they are well documented.

When you are auditing, you can chose to audit by session or by access. If you audit by session, you create a single record for each DISTINCT action a user takes. If you audit by access, you create an audit record for every occurrence of an action.

FGA

And now, on to FGA. As a refresher, FGA stands for Fine Grained Audit. FGA is context auditing, which is auditing based on the values in columns. FGA is implemented a lot like FGAC, which I discussed above.

You define a policy,

```
BEGIN
  DBMS_FGA.ADD_POLICY(
        object_schema => 'SCOTT',
        object_name => 'EMP',
        policy_name => 'scott_emp_sal',
        audit_condition => 'sal> 2500',
        statement_types => 'insert,update,delete,select');
END;
/
```

Any access on the EMP table that accesses a record with a SAL > 2500 will generate an audit record in the DBA_FGA_AUDIT_TRAIL view. So, this command by SCOTT:

```
SCOTT-SQL> select ename, job, sal
        2 from emp
        3 where ename = 'TURNER';

ENAME      JOB       SAL
---------- --------- ----------
TURNER     SALESMAN  1500
```

Generates no audit record, while the following query does generate an audit record:

```
SQL> select ename, job, sal
```

```
  2 from emp
  3 where ename = 'FORD';

ENAME      JOB       SAL
---------- --------- ----------
FORD       ANALYST   3000

SYS-SQL> select substr(db_user,1,15) username,
      2 substr(object_name,1,10) objname,
      3 substr(policy_name,1,15) policyname,
      4 statement_type
      5 from dba_fga_audit_trail
      6 /

USERNAME        OBJNAME    POLICYNAME      STATEME
--------------- ---------- --------------- -------
SCOTT           EMP        SCOTT_EMP_SAL   SELECT
```

The reason this generated a statement and the previous did not is based on the policy. The EMP with a SAL > 2500 generated an audit record. You do not have to access the policy column. The ploicy is evaluated regardless of what columns are accessed.

Optionally, when you define a policy, you can include a stored procedure. This procedure can then store additional information from the environment or generate an alert. The procedure is standard PL/SQL.

Trigger Based Audit

Finally, on to trigger based auditing. A trigger is a stored procedure that is tied to an object. An insert trigger on a table would execute every time an insert happened to that table. If we wanted to audit the new and old values of SAL in the emp table, we would define an audit table and a trigger on EMP.

```
SCOTT-SQL> CREATE TABLE emp_aud (
        2 empno number,
        3 old_sal number,
        4 new_sal number )
        5 /

Table created.

SCOTT-SQL> CREATE OR REPLACE TRIGGER emp_aud_trg
2   BEFORE UPDATE OF SAL ON EMP
3   FOR EACH ROW
4 BEGIN
5   INSERT INTO EMP_AUD
6     (EMPNO, OLD_SAL, NEW_SAL)
7     VALUES (:NEW.EMPNO, :OLD.SAL, :NEW.SAL);
8 END;
9 /
Trigger created.

SCOTT-SQL> select * from emp_aud;
```

```
no rows selected

SCOTT-SQL> insert into emp (empno,sal)
        2 values (-1, 400);

1 row created.

SCOTT-SQL> select * from emp_aud;

no rows selected

SCOTT-SQL> update emp
        2 set deptno = 10
        3 where empno = - 1;

1 row updated.

SCOTT-SQL> select * from emp_aud;

no rows selected

SCOTT-SQL> update emp
        2 set sal = 1500
        3 where empno = -1;

1 row updated.

SCOTT-SQL> select * from emp_aud;
EMPNO      OLD_SAL    NEW_SAL
---------- ---------- ----------
-1         400        1500

SCOTT-SQL>
```

You should not grant any permissions on the audit table to anyone. Only SCOTT would have permission to view or change the audit table. By granting permissions to access the EMP table, you are implicitly granting permissions on the trigger.

That's it. As you can see, standard audit requires no code and very little administration. FGA requires some code if you need to take special action but requires very little administration. Triggers require a lot of code and a lot of administration (whenever an object changes). When you can, opt for standard audit. When that doesn't work, try FGA. Finally, use triggers when that is the only way to meet your requirement.

Best Practices

Best Practice #1: Include the audit team at design time. Many times, there seems to be a confrontational relationship between IT and Audit that I don't really understand. In the end, you MUST meet your audit requirements. Deal with those requirements at design time and you will have no problems. Deal with it after implementation and suffer pain. Seems like a no brainer to me.

Best Practice #2: Plan for security at design time. Like audit, post-implementation security is painful and will usually require rework.

Best Practice #3: Only audit what needs to be audited. There IS overhead with auditing. If you need to audit, then you need to audit. Listen to your audit team, implement the auditing requirements identified and create a plan to cleanup audit trails on a regular basis.

Best Practice #4: In your production instances, secure everything. Lock down the database. No one gets access to anything without a reason. Start from here and work outwards. But don't get crazy. It is possible to go overboard. Don't treat your development and test databases like production.

Best Practice #5: Protect the audit trail. If you store the audit in the database, audit the audit trail. If you store it in the OS, protect the file system. In addition, if you use triggers, do not grant to the audit table, audit select, update and delete on the audit table and put triggers on the audit tables preventing update and delete. Also, use the parameter AUDIT_SYS_OPERATIONS = TRUE to force an OS audit for all activity by the DBAs when that's required. DBAs are normally exempt from fine-grained access and from auditing.

Best Practice #6: In an outward facing application with thousands of user, you have very unique security requirements. I've found that it's best to implement security for these through stored procedures. An application user, shared by many and authenticated by proxy, would have access to a very specific set of procedures. Those procedures perform any database activity required and return result sets for viewing.

Best Practice #7: For an inward facing application, and those with a finite number of users, every user should have their own database account. I have heard all of the arguments for sharing IDs and having the application maintain the users and they are all a load of, putting this politely, manure. Put security in the database, implement it once and sleep soundly at night.

Best Practice #8: Implement the user creation and role granting/revocation process through stored procedures. Give access to these procedures to the application administrator. That will allow creating users with consistent access across an application with out having to grant these powerful permissions directly. It also allows the DBA to offload basic application security.

Best Practice #9: Never grant direct access to anyone in production. Always grant access through roles. The only user that needs direct access is the schema that needs to compile procedures and views and that user should be locked once the application is compiled.

Best Practice #10: Use secure roles. Secure roles are roles that use a specified stored procedure to be enabled. That is, when the application says set this role for this user, it will execute a pre-defined stored procedure to validate that this person really does have access to this role, in this context. Think of secure roles as Fine Grained Roles (I just made up that term so only use it in this context). It's a procedurally implemented role.

Best Practice #11: Use fine-grained access when data needs to be secured at the column level. Use column masking to hide sensitive data.

Best Practice #12: Create and enforce a password policy. Oracle has the ability to execute a stored procedure to verify that a user's password meets your policy requirements. Oracle also maintains a password history and you can define how soon a password can be reused.

Best Practice #13: Use account locking. Set a number of retries for failing a password and then lock the account. You can set it to automatically unlock after a predetermined amount of time or force the user to call an administrator. This is to prevent brute force access.

Best Practice #14: Encrypt when needed but only when needed. If you store credit card numbers, bank account information, social security numbers, etc, you may want to encrypt that data. Oracle provides robust encryption in the form of the DBMS_CRYPTO package and Transparent Data Encryption.

I think that covers most of the best practices. I'm sure I skipped some things and glossed over others. These best practices are enough to get started.

I've kept this as brief as I could. I hope I made it plain how easy it is to meet your Audit requirements in Oracle. If you have any questions or comments, please email me at lewisc@acm.org.

If you would like to read more of the details on Oracle and Security, Oracle maintains a document, Oracle Database Security Guide, describing all of this functionality in great detail. Oracle also provides all of it's documentation, for free, on the Oracle Technology Network (otn.oracle.com).

Answers: Security and Garbage Collection

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006871.asp

Posted 12/12/2005

It was a very busy week and weekend for me so instead of finishing the database links article I'm writing, I thought I would try to answer some questions that have been posted to some of my past articles.

The questions are mostly left as I received them. I have edited for format and spelling (where I thought the misspelling was unintentional).

I'll start with Oracle Security: A Definition in Plain English.

From Rakesh Shah on 4/8/2005:

Hi

This is amazing article. I would like to learn more about this.

Thankx & Regards

Rakesh

My Reply:

Rakesh,

Thank you for the good words. If you would like to read the entire set of security articles I wrote, there are four entries, I've edited them as a white paper. It's formatted a little bit differently; part 3 is at the end. Anyway, you can get Oracle 10g Security and Audit here on ItToolbox.com.

Next up, on the same article is Rami ALSAHHAR on12/10/2005:

Hi,

Thanks for the information.

I was looking for information about Privacy and security for DBMS, one feature of Oracle is VPD. Still I am not satisfied.

Currently , I have a research on Hippocratic Databases to preserve the privacy. I need to know the latest work on this area. It is not how to protect DB or limit access control, it is about how to preserve privacy in DBMS.

Any Idea?

Thanks,

Rami

My Reply:

Rami, Thanks for the post. That's a great question and I see the point you're trying to make. Access Control and VPD keeps people out, but you want have the data encrypted even if the still get in (or get access to a backup tape or something like that) . The answer, as usual, depends on database version and exactly what you're trying to accomplish.

Since you mention being interested in the "latest work", I'll assume 10gR2. And since you mention a "Hippocratic Database", I'll assume you have sensitive information in specific columns but not everything needs to encrypted.

This is a timely question for me as I'm going through this exact same scenario at work. There are several areas to be concerned with. The first is the difference between data at rest and data in flight.

Data in flight is when the data is moving from point a to point b. If someone has a sniffer on your network and they pick up your data, they've gotten your "data in flight". I won't get into data in flight as that is a huge area and it's very dependent on the type of network (LAN, WAN, Wireless, etc), the type of network hardware (from what I hear, CISCO has some very nice hardware for this) and, honestly, I am not a network engineer. PGP seems like a good solution to me. That's really dependent on who's waiting at the other end though.

Data at rest is something I can address. In 10g, we have the package DBMS_CRYPTO. I mentioned this package briefly in my article on Binary XML. This package has functions to encrypt any data. If you have to encrypt XMLType, clobs, blobs, large varchar2 columns or key data (PK, FK, etc), I would recommend using DBMS_CRYPTO. DBMS_CRYPTO implements several standard encryption schemes. You can view the documentation here. Arup Nanda has a great article on DBMS_CRYPTO on otn.

Another way to encrypt your data in 10gR2 is Transparent Data Encryption (TDE). TDE will encrypt your data on the fly; there's no need to write any code. All of the encryption and decryption are handled by the Oracle server. I'm just starting to get into TDE. So far, it looks like it will work well for our needs. You can read another article by Arup Nanda (this guy is prolific) on otn. His article, Transparent Data Encryption was my first in depth reading on the topic.

A final note on encryption, Oracle has added encryption to RMAN. I don't have any good links for that feature but you can hit the documentation at tahiti.oracle.com or search on otn.

If you have any specific questions, or would like to see a prototype along these lines, post and let me know.

My final question is from my entry, Oracle Collections: A Definition in Plain English Part 2.

From George Catrombon on 11/23/2005:

One topic that never seems to be discussed when using constructors to create nested tables, is the cleanup of memory afterward. Unless PL/SQL uses a garbage collection scheme like Java or C# (which releases memory when a variable goes out of scope), the nested tables created by constructors will reside memory indefinitely. I don't recall that PL/SQL does this. My take is that you have to call the Delete() method to remove the elements of the nested table, at which point Oracle will de-allocate the memory when a full page of memory taken up by the nested table is freed up. Otherwise, I would think that all the memory would eventually be used up on the server, and crash the machine. I think that a discussion is badly needed on how Oracle uses memory when constructors are called over and over to create nested tables (collections) in PL/SQL packages/stored procedures.

My Reply:

George, I have not had time to research this yet but wanted to comment. You mention oracle deallocating pages of memory after calling delete. Did you read that somewhere? I really can't find much about this at all.

I believe scope does determine the lifetime of the memory used. When a stored procedure ends (i.e. goes out of scope), the memory is made available and will be reused if needed. It's possible to keep growing a collection until you run out of memory but even that will not crash the machine or the database. I can't remember the error off the top of my head, I don't think it's ora-04031, but you can only grab so much memory.

But, I'm pretty sure (and that means that this is an opinion, not a fact), that the lifetime of the memory is based on scope. Oracle doesn't necessarily release the memory, but it does make it available for reuse. That would also mean that a package variable would have a session as it's scope. In this case, manually freeing up the memory would be beneficial.

*As an aside, if you use nested tables in a SQL statement, I **believe** the scope for the memory is the lifetime of the transaction.*

If anyone out there has a link to some documentation about this topic, please post it. I for one am very interested. I'd love to hear your opinions also.

Ghost Town, New Orleans East, LA – Middle of the day, December 23rd, 2005

Chapter 10 - Leadership/Management

Leadership and learning are indispensable to each other. - John F. Kennedy

TOTD, What is a DBA?

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/003823.asp

Posted 4/12/2005

I was reading a newsgroup discussion and the topic of DBAs and their usefulness/trivialization came up (yet again). This topic seems to show up at one time or another on every Oracle discussion board.

Whether I agree or not with any of the things said in the discussion doesn't really matter. It got me thinking about the topic and I have a different perspective to the question. My personal soapbox on this issue is the definition of DBA. Are DBAs being trivialized in the enterprise? If they are, it's because the definition of that 3-letter abbreviation has become the garbage can of "all things Oracle".

First off, what does DBA stand for? Database Analyst? Database Administrator? Dumb Buffoonish Arrogance? Dippy Bippy Aiieee? Ok the last one is just silly, but I think I make my point. heh

Here are a few profound truths of DBAness:

- Putting the text "DBA" after a name does not impart instant knowledge of all things Oracle
- Not every DBA can design an application
- Not every DBA can design a database
- Not every DBA is happy administering a database
- Some DBAs ony want to administer the database

All of the above depend on what kind of DBA a person may be.

I can think of three very different kinds of DBAs right off the top of my head:

- Application Database Analyst - Should be able to do data modeling, application design, interfacing with project teams, etc. This person is concerned with building applications.
- Production Database Administrator - Guru on troubleshooting, performance guru, security, backup/recovery, etc. This person is concerned with getting the best performance and most reliable service from all production instances.
- Enterprise Database Analyst - Architecture, Standards, Oracle communications infrastructure and topography (replication, database links, etc), maintaining the overall enterprise architecture including number of production/test/development instances and how they're used. This person is concerned with the future direction and configuration management of the environment.

In any enterprise setting there will usually be some overlap between any of these. In addition, each of those above can be further split into specialties: data architect, data steward, modeler, application user administration, tuning, etc.

Can a single DBA do it all? Sure. I worked with a DBA when I was in New Orleans who could do all of that and develop applications on the side. Can you imagine having Thomas Kyte as your DBA? That would rock.

Can most people do it all? In my experience, not really and when they try, or when management forces them, they do it all badly.

So where am I going with this? No where really. Just whining. But I do wish the industry would try to standardize on some definitions and job descriptions.

When I see a job ad for DBA, at least half the time, I'm pretty sure they're either looking for a senior developer, data modeler or an application DBA but they'll say they need "x years as a production DBA". A person that is a production DBA probably won't be good in that position. And the same goes for an application DBA in a production DBA slot. They are completely different goals with completely different skill sets.

And I'm not even going to get into DBAs who are supposed to be the database police, whether through personal inclination or through management direction. Building an antagonistic environment is never a good idea.

I'm not sure how to change the perception that DBA is a single job function that can do everything. But if people who weren't "inside" the database could really understand just what it is that DBAs do, there would probably be less discussion of DBA trivialization. I guess this boils down, like so many other issues, to communication. Those of us in the industry need to better communicate to those in management and the end user community.

Is this an Oracle phenomenon or is it like this for all databases?

I'll step down off my soapbox now. How do you define DBA?

TOTD, What is a developer?

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/003854.asp

Posted 4/14/2005

What is a developer? Strange question, do you think? When I ask what is a DBA, everyone kind of shakes their head and agrees that DBAs have it bad. Pulled in twenty different directions and expected to know everything. But developers? Aren't they just coders? As usual, my topic will be from an Oracle point of view.

Before I get into today's entry though, I wanted to say I know I promised part 3 of the security definition. It's coming. This has been a very busy week. I'm doing some really cool things at work with analytics and it's just sucking up my brainpower and my time. I also just finished an article for the Oracle Topic I recently started at Suite101. If you know anyone who would like to learn oracle, please send them to my Oracle Topic.

And now, back on target:

I was primarily a contractor for many years so I have been in many environments. Every environment was different and every environment expected different things from their developers. In some places, a developer could not even create a table in the development environment. In others, they are expected to model the application, build the schema and deploy the application to production, giving it to the DBAs only after it's live. And of course, others were between those two extremes.

If you read my entry the other day, What is a DBA?, you know I talked about the various types of DBAs and some of their responsibilities. Today, I just want to point out a few things to be aware of about your development staff.

- To some developers, backend development is the holy grail of development.
- To some developers, backend development is the hellhole of development.
- Some developers love forms and are artists at creating useable front-ends.
- Some developers hate forms and can only create tele-type front-ends.
- Some developers can perform data modeling in their sleep and will design the best application you've ever seen.
- Some developers can't read an ERD.
- As hard as it is to believe, some developers WANT to be report developers.

My only goal with this entry is to remind you that if you have responsibility for hiring developers, learn something about what the real requirements of the position will be and find someone who fits (i.e. has the experience and WANTS to do that).

If you're not hiring Forms developers, why ask for 3 years of Developer 2000? I do see those ADs. If you're expecting the developer to design applications in addition to coding, make sure they can do that. Not everyone can. Not everyone wants to.

And if while hiring developers, you tell them they will be designing applications and later decide that only DBAs are qualified to perform that function, don't be surprised if they aren't happy about it.

If you're managing Oracle developers, remember that some developers are coders. Some do more. Some can do interfaces. Some can't. Remember that like DBAs, you need to play to an employee's strengths. Ask them on occasion to step outside their box, but don't expect them to be an expert when they do. And don't expect all of them to jump for joy when you do. They might like a new challenge; but they might not be good at it. At least not the first time.

To me, the most important thing I can say about managing developers is don't assume that you know what your developers are thinking or what they want to do. Ask them. Ask them often. This is one more area where open communication will make everyone's life a lot less stressful.

My next long article after the Security part 3 article and will be on building a dynamic Oracle development team. Who do you need, when do you need them and how do you need to use them.

Guru Wannabees

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/003924.asp

Posted 4/20/2005

Today I'm talking about something that I run into every now and then and really gets on my nerves. I call these people, Guru Wannabees. They really, really wanna be a guru. Because of their actions, and many times due to either a lack of effort or a lack of skill, they seldom rise above mediocrity.

As a project manager and/or team lead, I have at times ended up with Guru Wananbees (GW) on my team. Some of the symptoms that you have one of these on your group is a person who will:

- wait until the last minute to spring major issues on you
- will not commit to paper and are always at 95% complete, they have it all figured out but it must remain in their head and not on paper
- won't communicate with other team members, they must have a third party to intervene (probably you) or will only speak when directed to (usually at status meetings)

A GW impacts projects on several levels. The first is the risk factor. Unless you're creating something very minor, your project already carries risk. If one of your team members is a GW, then you don't really have a handle on your project timeline (last minute issues) and probably don't have a complete view of the technical details (will not document). The inability to communicate, and that means sharing ideas, puts you at risk of a less than optimal solution.

A GW also hurts project morale because if you notice it, you can bet others have also. This is especially an issue if you have a team member with a reputation as a GW who is never brought to task for it.

Why do GWs feel the need to act this way? For a lot of the time, I think it's the hero syndrome. To be the hero, there must first be a crisis. The GW can then ride out on a white horse and save the day. If they are sharing ideas and communicating, the likelihood of a crisis is minimized. Remember, after the battle, there's usually only one hero and a bunch of bodies. Most of your team is probably communicating and proactively finding and fixing issues. They are using their skills to succeed instead of hiding and/or creating problems.

Thinking of it that way, a GW is actively trying to sabotage the project. Not to actually make it unsuccessful but to impact it enough to make it late or a bit dirtier than it needs to be. These people want nothing more than to be considered a guru are forgetting that people aren't climbing the mountain to sit at the foot of a prima donna, they want sit at the foot of the master who shares his knowledge and wisdom.

Now, the hard question is what to do about your GW. Sometimes I hate the fact that communication seems to fix most ills. What I would like to do with GWs is kick them to the curb. But that's not always the best solution. It's been my experience as an employee and team member that many project manager and supervisors don't deal with the GWs. If you chose to ignore a problem, it won't just go away. And when dealing with personalities it will usually get worse and possibly spread to other people.

Therefore, I think the first step is setting guidelines. If the GW is a junior member of the team, pair them with a senior team member. They don't even have to have the same technical skills. Make it explicit that the senior person will be guiding their work. Try to pick a senior person that has the qualities you want on your team.

If it is a senior person, make it clear what the expectations are for senior team members. If someone is consistently rude to other team members, you would stop it immediately. The same is true for a GW.

Sit them down and make sure you are specific in the goals and requirements you are setting. Make sure they understand what specific issues you have with their behavior.

If the GW's issue is waiting for the last minute to spring issues on you, it might be too late for any of this on your current project. However, most GWs have a track record and in my opinion, past performance is an indicator of future performance. They also tend to spring smaller issues earlier on. As the project moves down the timeline, the issues get either larger or more critical. Sometimes these are legitimate, so again the track record is important. Good documentation helps here (see below). If possible, pair them with a junior person to mentor. Tell them you expect them to show this person, in detail, how to really do this job. Sometimes that is enough to kick in the hero complex and they want to get everything right.

What to do if your GW won't commit to paper or won't commit to a time estimate? This is always a fun one. In this case, I think the GW feels that if it's on paper, they're replaceable. It's somewhat hard to explain to a person with this viewpoint that they're infinitely more replaceable by being unreliable than they would be by providing good designs and estimates.

Start with your expectations. If a person doesn't know what's expected, they can't meet the expectations. Set deadlines for deliverables. It's almost like a mini-project. Let them know when you expect all or a portion of the deliverable and hold them accountable. If they don't try to meet the deadlines, you have a discipline problem not a GW.

Once you've set the expectations, either you should get results or you'll hear that they just don't have enough time. If they don't have time, make them explain how their time is being used. If they are only available for your project 30% of the time, you'll have to adjust your expectations of them. If the person is available to you 100% and you believe they should have enough time, it's probably a time management issue. Ask them to keep a log by the quarter or half hour for a few days and track everything they do during the course of the day. One or the other of you will probably be surprised at where the time goes. From there, it should just be a matter of prioritization.

On the subject of non-communication, it's worthwhile to see if it's a specific person or people that your GW has an issue with. As professionals, people should be able to work around any personal issues they might have. However, what I have found to be the most common reason for this is the GW feels that someone with less skills is in a more senior position. The solution here is not to address their personal issues but to address the lack of communication. I've learned the hard way that you can't magically fix this conflict. Let the GW know that you expect them to set aside any differences and communicate as needed.

If the lack of communication is with the team in general, it's up to you to include them in the team. Specifically ask their opinion in meetings. Make sure they're included in the decision making process. This might not be a GW in action. Not everyone is a natural born team player and it's the lead's job to make sure the team is not dysfunctional. Talk to them alone and let them know that you value their opinions and ideas.

In all of these cases, communicate your concerns with that person. This comes back to my saying of, if a person doesn't know what's expected, they can't meet the expectations. What seems obvious to you may not be so obvious to someone else. I have made assumptions that I knew what a team member wanted or was thinking or was motivated by and have been wrong just about every time. Never assume. It is so easy to just ask. The conversation should not be confrontational. If it is, you've waited too long.

And finally, if any of these issues are tied to defiance, you have a discipline problem. If you communicate and make sure they understand your expectations, and you still can't work it out, you have a discipline problem. As a last resort, follow your company's guidelines for disciplinary action. But before you do that, remember that as the lead, you have a tremendous responsibility to try to make that not happen.

That's my soapbox of the day. I'm working on a part 4 to my security series. I decided to do a complete walkthrough of the auditing features as I did in part 2 for the security features.

How to interview an Oracle Developer

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004054.asp

Posted 5/3/2005

It looks like the job market for Oracle is opening up. I'm seeing many of my friends find new jobs and I'm doing a lot of interviewing at work. I hope this is a trend that continues. In this entry, I share my method for interviewing Oracle resources and provide some sample interview questions and answers for developers.

Interviewing anyone can be difficult. Interviewing technical resources is very difficult. To me, probably the hardest thing is pinpointing exactly what you want this person to do. What, exactly, will this person be doing in their day-to-day job? It's easy to say having a good job description makes it easier but in many cases, employers have a generic template when looking for someone.

First off, is there a difference between a developer and a programmer? I think there is. A programmer is a coder. I don't mean that as a bad thing. Every project needs coders. A developer should be part analyst and part coder. A developer should be able to handle requirements gathering through implementation. Every project should have at least one good developer. In your interview, you should distinguish between the two.

If you're looking for an Oracle resource, there is a huge possible range of knowledge. Does a developer code for the front end? If so, which tools? Forms, Reports, HTML DB? I hope that you aren't looking for Java because wouldn't that make them a java programmer and not an Oracle developer?

Before you start to interview people, make sure you really know what you're looking for. It's not fair to the candidate to say you're looking for forms experience and then spend all of your time on advanced back end programming. And don't ask a backend coder the fine details of forms.

When I interview someone, I look for more than yes/no answers to my questions. In an interview, I expect a candidate to communicate with me. I'm looking for a comfort level. If the person says they don't know a particular topic, that's acceptable. If they fumble and make something up, that's a problem to me. I don't look for textbook definitions. I want to know they understand what I'm asking and what they're answering. I also don't believe in tricky interviews. What's the point?

Regardless of the exact position, any Oracle resource, including a DBA, should know some basic things about SQL and PL/SQL. Some sample questions I ask are:

For Basic SQL:

- How do you convert a date to a string? To_char. A bonus would be that they always include a format mask.

- What is an aggregate function? I'm looking for "grouping", sums or counts, etc.
- What is an interval? Specifies a period of time.
- What is a nested subquery? A subquery in a where clause.
- What is the dual table? A single row table provided by oracle for selecting values and expressions.

For Basic PL/SQL:

- Describe the block structure of PLSQL. Declaration, Begin, exception, end.
- What is an anonymous block? Unnamed PL/SQL block.
- What is a PL/SQL collection? PL/SQL Table, Varray, PL/SQL Array, etc.
- What is the difference between an explicit cursor and a select into. You might get something about performance but that's a myth. An explicit cursor is just more typing. A cursor for loop would be used to return more than a single row.
- Why would you choose to use a package versus straight procedures and functions? I look for maintenance, grouping logical functionality, dependency management, etc. I want to believe that they believe using packages is a "good thing".

These are pretty basic questions. If I don't get a warm fuzzy from these, and they are 100% answerable by anyone with some real experience, then the person goes no further.

So, where do you go after the basics? That really depends on what you're looking for. If you are hiring a Java coder to work with your Oracle group or you're looking for a DBA, you might end the coding part here. You would expect a DBA to know more but I would move on to administrative questions. You might also stop here if you're looking for a junior developer to train.

If you're looking for a senior PL/SQL coder type, you will want to go deeper. You need to remember to ask specific questions about a person's background and forms developers will have different experience than a back-end developer. But either should have a good grasp of advanced topics.

The hard part is that there are so many advanced topics; it's hard to know what to ask. You need to tailor it for your environment. If you use a lot of AQ, ask AQ questions. If you're very OO, ask OO questions.

Here are some more advanced, but still generic questions:

For Advanced SQL:

- What is the difference between an aggregate and an analytic function? I'm looking for them knowing that a sum aggregate (or any other aggregate function) will return one row for a group and a sum analytic will return one result for each row in the group. If they mention the "Window", they get a bonus point. ;-)
- How do you create a hierarchical query? Connect by.

- How would you generate XML from a query? The answer here is "A lot of different ways". They should know that there are SQL functions: XMLELEMENT, XMLFOREST, etc and PL/SQL functions: DBMS_XMLGEN, DBMS_XMLQUERY, etc.
- What do you need before implementing a member function? You need to create a type.
- How do you tune a query? I'm looking for a discussion of autotrace and/or explain plan. Ask them what they're looking for in a plan. This should not be a single sentence. Look for a comfort level.

For Somewhat Advanced PL/SQL:

- What is the default value of a boolean? NULL. This is somewhat tricky but apparently there are languages that default boolean to false. A PL/SQL developer needs to know all variables default to NULL.
- Why is using implicit conversions a poor programming practice? For dates, you must ASSUME that the default date format will always be the same (and it won't be). In some cases, implicit conversion is slower. I want to feel like they don't believe writing to_char or to_number is more work than it's worth. BTW, this also applies to SQL.
- How can you tell if an UPDATE updated no rows? SQL%NOTFOUND.
- How can you tell if a SELECT returned no rows. NO_DATA_FOUND exception.
- How do you run Native Dynamic SQL? Execute immediate.
- What is an autonomous transaction? Identified by pragma autonomous. A child transaction separate from the parent that MUST be committed or rolled back.

Those are some items that should also give you a warm fuzzy. If a person makes it to here, you can ask the questions specific to your organization, i.e. the AQ, LOB, Forms, HTML DB, etc.

At this point I usually ask the candidate to explain specific statements on the resume. If they say they tuned queries or improved performance, I say how? What did you do? What tools did you use?

That's my interviewing method. I hope that helps you get the best people for your organization.

Dont Step in the Leadership

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004113.asp

Posted 5/9/2005

My apologies to Scott Adams and Dilbert. Don't Step In The Leadership is the title of one of Scott's books. I have the entire Dilbert collection including the more esoteric Dilbert Principle, Joy of Work and Way of the Weasel. I heartily recommend all of the Dilbert books.

I was reading a blog here on ittoolbox the other day and the title phrase jumped into my head. It doesn't really matter which blog it was or why it made me think of Dilbert but I decided to address the subject of, and my thoughts on, leading and being a leader.

Scott Adams has an opinion on leadership here. Got to respect that.

Being a leader is not something I ponder often. I normally just try to do the best I can at whatever it is I'm doing. My first question to myself was what did I mean by leading. I don't mean that when I was a child I wanted to grow and be a middle manager. heh What ad is that from? I don't remember but it's a very funny commercial.

When I say leader, I mean several things. Most importantly, I mean someone out front. Not in front in the sense of bleeding edge technology. You can't always choose your projects or job. A leader is not necessarily the boss or a manager. A leader strives to be the one to follow. A technical leader, and that's how I think of myself, strives to be at the top of their profession. I may never reach the pinnacle but that's what I strive for. That goal requires research, reading, etc. It's a huge time investment. It also requires a certain amount of discipline.

Regardless of what type of leader a person is, or wants to be, I think a leader must maintain certain qualities. It's no different in the blogosphere. Below are some qualities that I try to maintain, whether I'm in my office, in a classroom, or writing my blog.

Credible - Research your facts. Facts without proof is fluff. If you have an opinion, state it as an opinion. When I write my technical entries, I double check my facts. That doesn't mean I'm not going to have mistakes but my mistakes should be rare enough that I can be a credible source of information. For a blogger, pointless tirades and tabloid headlines destroy credibility. It's possible to be humorous and/or creative without being exploitative. A blog headline can be written to grab someone's attention without slapping them in the face.

Open Minded - You don't have to agree with every supposition posited. You should be open to change, though. That latest and greatest fad might actually be a good thing. When you decide that you know so much that you no longer have to listen to the other side, you stop being a leader and become a dictator. Even when I don't particularly believe in something, I try to see it from the proponents viewpoint. You lead when you use all available knowledge to make an intelligent decision. A leader avoids knee jerk decisions.

Critical - That latest and greatest fad might be great but it might also be a boatload of crap. While I'm trying to see someone's viewpoint, I'm maintaining a healthy dose of reality. Not everything that glitters is gold and not everything written is true. Balance an open mind with critical thinking. Don't jump to conclusions; either for or against.

An Example - As in, be one. A leader is someone to be followed. A leader that wanders through rumors and tabloid country will quickly be abandoned for more reliable leadership. A leader should display integrity and reliability. Being dependable is nearly as important as being credible. If your actions can't be held up as an example to emulate, you are not a leader.

Professional - Act like a professional. Instigating conflict to make yourself look better is as bad as knocking down others for the same reason. Professionals communicate; they do not confront others, and they don't insult others, for personal gain.

Those are my thoughts. I apologize for the soapbox. I will climb down now.

Besides being very enjoyable, one of the things that I have discovered while writing my blog here at ittoolbox is that it makes me clarify my thoughts. I want to be as concise and clear as I can be when I write an entry and I can't be clear on a topic if that topic is not clear in my head.

As I thought about leadership in general, it occurred to me that I had some very specific thoughts about what makes a good leader. The above thoughts are just my opinions. I know they won't start a new fad in management but I wanted to say them. If you've made it this far, thanks for reading.

As ever and alwys, if you have comments, critical or not, please post.

And remember, don't step in the leadership.

Are DBAs needed anymore?

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006034.asp

Posted 5/17/2005

I just read this article about a debate at IOUG in Orlando a couple of weeks ago and thought I would pass it along.

It really got me thinking this morning. Nothing like a cup of coffee and a kick in the head to get you started!

Are DBAs still needed? Very BIG yes! And not just because that's my title. heh

Oracle 10g has added a lot of new features and I like a lot of them. At this point though DBAs are still required for many things, installation and upgrades being a part of that. OEM will look for patches for you and the installs are almost no brainers but there are still a few gotchas out there.

It's also nice to have the automated management and tuning features but right now it's like dumb AI in a game. Good when it works but a major pain when it doesn't. In a few more minor versions, maybe good enough? We'll have to see.

But after the database is fully automated, say 11q, will DBAs be needed? I think we'll get to the point where management really needs to distinguish between a data analyst and a data architect and a data ???. See my write-up TOTD, What is a DBA?.

Right now there is still a place for script monkey DBAs. In the future I expect that kind of job to either go away or be relegated to a database operator (remember the DBO?). In earlier versions, the concept of a DBO was just silly. You had to have so much internal knowledge of Oracle that you were a DBA by the time you became a DBO. In the near future I see that as being a valid trade skill type career. People could use it as a springboard into bigger things or stay and become an uber DBO.

I think the current version is a great opportunity for developers who want to move into a more DBA type role. You still need to learn the architecture but the automated tools make getting a database up and running, that much easier. At this point in time, it is reasonable to expect a senior developer to install and configure a basic production database. And for those really senior developers who have some background with what is considered DBA territory, mainly configuration, it's even feasible to setup and maintain DSS or high volume OLTP systems.

What do DBAs need to do to stay relevant? The need for hard-core internals knowledge will not go away. There will always be a need for the person who can recover from problems not planned for (which means can't be programmed for) and for performance tuning when the tools just aren't doing it for you. There will also be the need for security and auditing gurus. Oracle makes both fairly simple to implement but I think you need a person to manage it. That may get passed on to a security or auditing group rather than the DBAs though.

But I think to be DBA today and tomorrow, you need to learn the business. **WHY** is your company doing **WHAT** it is doing? DBAs will need to make sitting down with, and understanding, the business users as a high a priority as they do recovery and tuning today.

DBAs will also need to expand their horizons. I find that most DBAs pick a subject and become an expert in that: RAC, performance, recovery, etc. I feel that in the future of automation, a generalist who understands the business will be much more valuable to a company.

Looking for a job?

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006034.asp

Posted 10/6/2005

A good friend and coworker of mine gave his resignation notice recently. He's going to be going to a great job that I think will challenge him and give him some super opportunities. I wish him the best!

That got me thinking about the job market. I'm doing a lot of interviews here at work both for contract help and permanent employees. I'm also being contacted by recruiters on a semi-regular basis. Some want to know what my needs are but many of them want a copy of my resume. That's always a nice feeling.

Anyway, while I was thinking about the job market, I started thinking about all of the job boards. I figure those web boards are the best indicator of the current market. So, I decided to make a completely unscientific survey of the job boards that I frequent to see how many oracle jobs I could find.

I queried DICE, Monster, CareerBuilder, Hotjobs and OracJobs. I'm sure you've heard of most of these. Oracjobs.com is the Oracle Job Network. HotJobs.com is the yahoo job board. The other three I'm sure you're familiar with.

I queried these 5 job boards for 5 different search criteria:

- oracle
- oracle dba
- oracle developer
- oracle java
- oracle xml

I knew the searches would have some overlap between them, i.e. oracle developer and oracle java.

I also queried them for 4 different locations:

- All Over (entire US)
- Southeast US (Al, Ar, Fl, Ga, Ms, La, Tn, SC, NC)
- Florida
- Tampa/St Pete/Bradenton, FL (that's where I live)

Some of the results below surprised me.

I'll run through each search but first some comments on the limitations of the boards themselves.

The big names, including hotjobs, are pretty much the same when it comes to usability. I think overall I like DICE best. If you are going to be doing a lot of searches like those that I did, Dice and HotJobs are the easiest to use. I don't think many people do massive searches like this but if you're a recruiter searching resumes, it might make a difference.

Some things I don't like:

- Oracjobs.com can only choose from lists, there are no entry boxes. You can't search on keywords, only on selections from the category (developer, dba, manager, etc) list. You can't enter locations, you can only choose from the locations list.
- Oracjobs.com will not let you restrict your search to a number of days back. You get all hits every time. From reading the site, I think they only keep 30 days worth of ads but I still would like to limit it.
- HotJobs and CareerBuilder cannot search for more than three states at a time and have no regional search.
- Hotjobs could only select two cities (with surrounding areas)
- When trying to search all of Florida using hotjobs, it kept putting Florida, NY in the location field. I finally picked Florida from the left hand list of locations and that worked.
- Monster does not show total counts over 1000.

For these searches, I limited the queries to jobs posted in the last 7 days. On Oracjobs, I just scrolled down and counted the number of jobs until I hit a date beyond the 7-day limit.

Oracle Anywhere in the US

	DICE	**Monster**	**HotJobs**	**CareerBuilder**	**OracJobs**
oracle	8662	> 1000	5989	2769	< 100
oracle dba	1120	275	458	392	9
oracle developer	3544	752	1493	1070	9
oracle java	3093	887	1520	1199	NA

oracle xml	1606	500	925	340	NA

The NA columns for OracJobs is because that board does not allow keyword searches.

There was an impressive number of oracle jobs listed and this is just in the last 7 days. I haven't seen numbers like since pre-Y2K.

I was surprised that HotJobs had more listings than CareerBuilder. DICE, Monster and CareerBuilder have always been the big three on my job searches. I will definitely integrate HotJobs into any searches in the future.

Oracle In The Southeast

	DICE	**Monster**	**HotJobs**	**CareerBuilder**	**OracJobs**
oracle	1012	412	NA	NA	18
oracle dba	142	37	NA	NA	3
oracle developer	438	85	NA	NA	4
oracle java	327	77	NA	NA	NA
oracle xml	184	61	NA	NA	NA

At neither CareerBuilder nor HotJobs could I find a way to do a regional search. Neither will allow you to enter more than 2 or 3 states and there was no regional option. That's the reason for the NAs.

Oracle in Florida

	DICE	**Monster**	**HotJobs**	**CareerBuilder**	**OracJobs**
oracle	309	155	134	133	15
oracle dba	54	7	12	19	3
oracle developer	121	30	43	46	5
oracle java	90	30	35	49	NA
oracle xml	56	37	29	13	NA

Comparing the numbers just in Florida against the Southeast, it looks like a good portion of Southeast jobs are in Florida.

Oracle in Tampa/St Pete/Bradenton

	DICE	**Monster**	**HotJobs**	**CareerBuilder**	**OracJobs**
oracle	89	52	40	41	NA
oracle dba	23	17	3	5	NA
oracle developer	36	47	11	16	NA
oracle java	28	47	10	16	NA
oracle xml	16	34	9	7	NA

This is where I live. Technically, I could have included Orlando but that's about an hour away while the three I did list are all with 45 minutes.

You cannot restrict Oracjobs to specific cities. State is the finest granularity it allows. That will account for the NAs in the final table above.

DICE and Monster are the larger of all but DICE seems to consistently have the largest number of listings.

So, what do I get out of these numbers? If you want to work with Oracle, being an Oracle developer is your best bet in sheer number of jobs. And with these numbers it looks like if you know Java on Oracle you're pretty much ahead of the game. XML is catching up with Java.

After running these searches, I thought it would probably have been interesting to run some for "oracle sql" or "oracle pl/sql". I wonder how much different the hits would have been? Maybe in the future.

Does this mean I'm going to be sending out my resume any time soon? Right now, in my job, the upsides outweigh the downsides. However, it is nice to know that things are looking up.

I'm not even sure what I would be looking for if I do leave my current job. Architect? Development Lead or Manager? Hard-core Oracle evangelist? I'm not sure. Based on the things I write about, what would you say my skill sets/interests are?

Help me out here.

Are you hiring now? What part of the country are you in and what kind of Oracle positions are you filling? Will you be hiring into 2006?

Are you looking for a job? What kind of job are you looking for and how is the hunt going?

My Philosophy of Interviewing

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006219.asp

Posted 10/20/2005

In May, I posted an entry, How to Interview an Oracle Developer. This was a popular entry back then and seems to keep coming up, even now. Yesterday I got a comment on it from a reader, rash. Based on that comment, I decided to take another look at the entry and clarify my philosophy on interviewing candidates for an oracle development position.

Rash's comment was:
Hello Lewis,

I've been working on oracle for the last 10 years. However, I leave a lot of thing to search/refer when I need. I have not attended interviews in a long time. Some of your questions are simple, but I couldn't have answered it immediately. thanks.

First of all, thanks for the comment rash!

And now on to my philosophy. In that entry I wrote:

When I interview someone, I look for more than yes/no answers to my questions. In an interview, I expect a candidate to communicate with me. I'm looking for a comfort level. If the person says they don't know a particular topic, that's acceptable. If they fumble and make something up, that's a problem to me. I don't look for textbook definitions. I want to know they understand what I'm asking and what they're answering. I also don't believe in tricky interviews. What's the point?

And that really, is my philosophy. I look for a comfort level. Let me ask you rash, even though you couldn't have answered some of the questions immediately, could you have spoken comfortably about each? Not knowing the answer to a question, especially one about syntax or some more obscure SQL command, is not a reason to write a person off. If you can't remember the syntax, say what it does or say how you used or even say that you haven't used it and can't remember exactly what it does but that should you need it you know where the SQL reference manual on OTN is. Anything is better than, "I have no clue."

Now having said that, in the entry I list some Basic SQL and PL/SQL questions. Those I would expect anyone who calls himself or herself a developer to know. The way the question is worded may need to change but the candidate should know the answer. I don't mind providing additional explanation of what I'm looking for if the question is not clear to the person, but I do expect a positive answer.

It's a different game with the questions I labeled advanced. Everyone's background is different. What is advanced to me may be very basic to someone else. Or what is advanced to me could seem arbitrarily obscure to someone else. Gear your questions around the skills you need for the position you're hiring for.

I mentioned that in these two paragraphs:

If you're looking for an Oracle resource, there is a huge possible range of knowledge. Does a developer code for the front end? If so, which tools? Forms, Reports, HTML DB? I hope that you aren't looking for Java because wouldn't that make them a java programmer and not an Oracle developer?

Before you start to interview people, make sure you really know what you're looking for. It's not fair to the candidate to say you're looking for forms experience and then spend all of your time on advanced back end programming. And don't ask a backend coder the fine details of forms.

Make sure you hire the people with the skills you're looking for. And tailor your interview questions to those skills.

Going back to my philosophy though, let me run through some possible answers to one of my questions to highlight what I look for.

Me: What do you need before implementing a member function?
Candidate: I don't know.
WRONG ANSWER. Did this person not understand the question or doesn't know about OO?

Me: What do you need before implementing a member function?
Candidate: I'm not sure what you're looking for. Can you expand on that?
Me: I'm looking for your knowledge on objects in Oracle.
Candidate: I haven't done that.
STILL WRONG ANSWER: Better than the previous but still lacking.

Me: What do you need before implementing a member function?
Candidate: I'm not sure what you're looking for. Can you expand on that?
Me: I'm looking for your knowledge on objects in Oracle.
Candidate: I haven't done OO in Oracle but I have read about types in the Object-Relational Developers Reference on otn. I'm not sure what a method is.
GOOD ANSWER: This person hasn't needed objects on their day to day job (I mean, really, who has?) but knows where to go for more information.

Me: What do you need before implementing a member function?
Candidate: I'm not sure what you're looking for. Can you expand on that?
Me: I'm looking for your knowledge on objects in Oracle.
Candidate: I haven't done OO in Oracle but I have read about it. Methods are code or something in a type.
GOOD ANSWER: This person has at least made some effort to read about it.

Me: What do you need before implementing a member function?
Candidate: I'm not sure what you're looking for. Can you expand on that?
Me: I'm looking for your knowledge on objects in Oracle.
Candidate: I've created some types in PL/SQL but have not gone as far as creating methods. I don't really like OO in Oracle.
REALLY GOOD ANSWER: This is a good knowledge and has an opinion about its usefulness.

So that is the kind of thing I look for. It's ok not to know the answer as long as you make sure you understand the question and you make me understand you have some initiative and desire to do the job you're being hired to do. I want to know that even if you haven't done something, you've made the effort to read and learn about it.

When you're interviewing, yes, no and I don't know are bad answers always. If I can't get more than three words out of you during the interview, what can I expect when I hire you? There's no need to be aggressive or the most outgoing person, I'm not either of those, but I do expect the interviewee to make sure I know how comfortable they will be in the job.

And that's about it for my interviewing philosophy. I'll leave you with one more example of the interview question:

Me: What do you need before implementing a member function?
Candidate: A type. I learned that from an Expert's Guide to Oracle on ITToolBox.com. That's the best blog in the world.
THE RIGHT ANSWER: This person should be hired immediately!

Lighten up! My take on the Oracle Expert Debate

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/003555.asp

Posted 4/22/2005

If you're involved with Oracle in a technical capacity, then you are familiar with Thomas Kyte and probably Don Burleson.

They have been having a very visible debate on asktom and on Don Burleson's site, Burleson Consulting. One example is in the forum. For other examples, search on Don's site's forum or on asktom.

I wouldn't be posting this entry at all but now that debate, and an ugly debate it has become at times, is spilling over to other sites that I frequent and that just got on my last nerve. I would post the link to the other site but you have to sign up and become a member to view the article and I don't want my blog to become an advertisement to other sites. ;-)

The gist of the argument started, I think anyway, about methodology. It then degenerated into who is an expert and who isn't. It's also sucked other very respectable professionals into it. See "Want to debate" on the asktom mainpage.

The article I mentioned above (the one on the other site) is ridiculous as is the entire argument about who can be an expert. I think the most ridiculous thing about it is the arrogance and pompousness of the author.

Here is a blurb from the article's headline: "How can an Oracle pro with a technical question separate the gurus from the poseurs? Don Burleson offers his opinion on the poor state of the Oracle "expert" community and how to find the wheat, not the chaff."

He has a right to post his opinions wherever he wants and I would never suggest otherwise. However, after reading that article, I have to exercise my right to write this entry. The saddest thing is that I own a couple of his books and don't in any way think he's stupid. But I have lost a lot of respect for him over this subject.

Thomas Kyte has written a couple of books, both of which I think are worth their weight in gold. I think he was tempted to get down and join the fray (that's just my belief from some of the entries on asktom and his new blog). But, as a professional, he seems to have had his say and dropped it. His site is probably the most valuable for Oracle professionals on the web after OTN and Metalink. I'm glad he chose the path he did.

So where does that leave us? With my opinion of course. What do I think of the argument? They started with the methodology of proving performance tuning (at least that's where I first noticed it). So, my opinion on that topic is that they both know their subject better than I do. I would try both in a test environment and chose the one that worked best. No egos, no opinions, just a matter of what works and what doesn't.

And what about the debate about what an expert is and who is entitled to that esteemed title? There are many sites with good oracle information. There are probably just as many with bad information. If you pick a site and follow its expert/scientist/guru/whatever blindly, you deserve what you get. If you read a tip and implement it without testing it, you deserve what you get. This is common sense people. It's not just an oracle argument. If you run sql server and blindly follow tips from the internet, you deserve what you get.

Who cares what people call themselves. If you want to know their credentials, ask them. Personally, I don't care. What I care about is consistently good information. I check facts against at least a second source and whenever I try a tip or method, I test it first. In my early days reading asktom, I tried to break his stuff and prove him wrong. I quit trying that after a while and just accept that he is an expert. I still test anything I get from his site though.

So what is an expert? Isn't that what Don's article is about? An expert is someone who has expertise in a subject. Check out this definition: http://selfknowledge.com/34107.htm

I'm pretty sure neither Don nor Thomas have been visiting my blog but I will use it as an example. My blog is titled: An Expert's Guide to Oracle Technology. Do I have any right to call myself an expert? First off, I didn't choose that title. ItToolbox did. I think my suggestion was something like "An Oracle Encyclopedia for Executives and Project Managers". You can see why they chose a title for me. heh

But, am I entitled to call myself an expert? According to the above definition, I am. I have expertise in a wide range of Oracle technology. I also have expertise in different areas like project management, supervisory, etc. I even have expertise in areas totally not related to oracle or even computers. I make a mean Gumbo! Are there people with a greater amount of expertise? Of course. When I can't figure something out I go to asktom.oracle.com. I work with people who have different areas of expertise that I consult with on a daily basis. That is how I increase my areas of expertise.

I'm also pretty sure I have expertise in areas of Oracle that Thomas and Don do not. We have had different professional experiences and histories and that is a good thing. How boring would it be if you only had a single source of information?

Should anyone follow me blindly? I sure hope no one does. When I say something in this blog is a fact, I research that and, if it's code related, like my recent security posts, I actually test everything I'm writing about to make sure it's true. It all worked on my computer, in my environment with my particular version of Oracle. You may get different mileage.

Now, when I say "I think" or "My opinion", that's different. I hope I'm making it plain that in that case, I'm just spouting my opinion or making a best guess.

And finally, does anyone really walk around wondering if the rest of the world thinks they are an expert? That's just sad a thought.

On an up note, in Don's article he suggests you research your guru by entering:
oracle "experts_name_here" (CV | C.V. | resume | experience | worked')
into google. I did it with my name and got some interesting results. I got different results when I included my middle initial. I got some hits from years ago. Ah, the old times.

Well, I think that's it from this expert. Until next time, remember to test. And floss. I'm off to get in touch with my inner guru.

Lakeview, La – December 24th, 2005

Chapter 11 - Conferences and Seminars

Think like a wise man but communicate in the language of the people. - William Butler Yeats

A day with Ralph Kimball, Part 1

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/003555.asp

Posted 3/21/2005

I had the opportunity to spend a day in a seminar with Ralph Kimball. If you don't know who that is, he is a guru of data warehousing. If you're involved in data warehouses, I hope you are at least familiar with his work. Currently in the industry there are two primary, competing warehousing methodologies, i.e. practically religions to some, Kimball vs. Inmon. I think that's kind of silly. A methodology is like a hammer or a drill; choose the best one for the job. If I absolutely have to pick, I'm in the Kimball camp. I have used Inmon's methodology in the past and that works but I find that Kimball's work applies better in my most recent projects, particularly when implementing enterprise bus architecture. Having said that, if I was on a project and the requirements made me think Inmon's methodology would work better, I wouldn't hesitate to do so. For that matter, just using flat, reporting tables is appropriate in the right circumstances. But I'm getting ahead of myself. Back to the seminar.

An extract from the marketing literature: Ralph Kimball is the founder of Kimball Group, a company that trains dimensional modeling and critiques data warehouses. He founded Red Brick, a super speedy warehouse RDBMS that was purchased by IBM. Before that he was a vice president with Metaphor Systems who sold data warehouse systems and before that he was at Xerox Parc where he co-invented the system that used a mouse and GUI, the Xerox Star Workstation. He also authored the Data Warehouse Toolkit, which in my opinion is indispensable for warehousing professionals.

This guy is rock solid. He doesn't just write books. He's been implementing these ideas for years and proving they do work in the real world. The clue I would like to get across is to follow Kimball's guidelines fully. If you only implement a piece of it, don't complain that Kimball's methods don't work. Combining star schemas and fully relational schemas or taking what you believe are short cuts to make things simpler will probably fail. Stick to the proven method. But that's my soapbox and I'll step down now.

When I got up that morning it was raining. It had been raining the evening before and my normally 35-minute or so commute became a 1-½ hour commute. The Marriott in downtown Tampa is twice as far from my house as is my employer. I almost decided not to go. The combination of rain, traffic and the fear that it would just be a marketing gimmick almost made me decide not to attend. At the last minute I decided that this might be a one-time opportunity so I drug myself out of bed and got on the road 2 hours before show time. I'm glad I made that decision.

The traffic was bad; as bad as I figured it would be. But it flowed and I didn't hit any wrecks. I managed to make it in 1 hour, which is good time for me. There were a lot of friendly people waving to each other with one finger but no road rage as far as I could tell. One of my co-workers who also attended started off about 10 minutes behind me. It took her a lot longer to arrive though. I must have left at the right time.

The downtown Marriott, Actually called the Tampa Marriott Waterside Hotel and Marina, is a NICE hotel. The meeting room was high enough up that the hallway outside had great views. The view over the water and the marina is amazing. If you're ever coming to Tampa, and someone else is paying, you should consider it. Here's the link: http://marriott.com/property/propertypage/TPAMC

The seminar was arranged, and I guess paid for, by Informatica, Sybase and Trillium Software. They put on a good spread. When I got there, an hour early, after I registered and got my badge, I had my choice of breakfast. The provided orange juice and coffee to drink and bagels, muffins, fruit, etc to eat. I give four stars for the munchy factor. Food is important to me. Food is my friend. I ate way more than I should have. I think I might have had one of each. Or maybe I had more than one.

My co-workers started showing up about ½ hour later at 8am. I passed the time reading the latest Dr Dobbs Journal and eating. I thought the seminar was supposed to start at 8:30 but it turned out that breakfast was supposed to start at 8:30. I was full by then.

Finally at 9, the informatica rep got the proceedings started. The schedule we were provided was:

```
8:30-9:00 Breakfast and Registration
9:00-10:30 Ralph Kimball
10:30-10:45 Break
10:45-12:00 Informatica, Sybase and Trillium
12:00-13:00 Lunch and Q&A with Ralph
13:00-14:30 Afternoon Workshop Part 2
14:30-14:45 Break
14:45-16:00 Afternoon Workshop Part 3
```

Not bad I thought. If Ralph had the majority of the day, it might be worthwhile after all. The rep introduced Ralph and Ralph got started. It turns out that the daylong seminar was based on his classes. Basically it was to give us a taste of what his classes provide. The taste was enough that I really, really want to take his classes. The classes aren't cheap though. The two I would like the most are Dimensional Modeling in Depth and Data Warehouse ETL in Depth. You can see his web site and info about he classes at: www.kimballgroup.com

Let me say that if you have already read and studied his book, the data warehouse lifecycle toolkit, this one-day seminar will not teach you anything new. Having him say it the way he does and having him present his experiences and opinions really solidifies what the book says. Having the opportunity to speak with him on breaks and at the Q&A is worth the parking fee anytime. He also said his classes are like boot camps where you are dealing with multiple real world scenarios and actually implementing data marts and doing ETL (in the ETL class).

The morning session covered the basic lifecycle of a warehouse, a bit about dimensional modeling and some definitions. It was the executive overview of the day. He made some very good points. One particular point is that he considers himself an engineer not a scientist. A scientist can ponder theory forever while an engineer delivers working products. I see myself in the same light. I like considering the theories and bouncing ideas around but at the end of the day, I apply theories. I like to design and build but then I want to move on to the next project, not coddle a particular project forever.

Another comment he made is that if you think after interviewing your users, you have defined all of the requirements, then what you really have is foofoo dust. I'm not sure if I spelled that correctly but I don't think it really matters. I WILL be using that one in a meeting sometime in the future.

His presentation was an analogy to the publishing industry. Like the publishing industry, the warehouse group collects input from a variety of sources and edits for quality. I think two of the most critical things he spoke about are that data warehouse professionals must maintain the end users trust. Data quality, timeliness and just being honest about expectations are paramount. And at the end of the day, success is measured by the happiness of your end users. If it's fast but nobody uses it, it's not a success. If it's too slow to be usable or the data is too stale or the interface, i.e. schema, is too confusing, there are no excuses.

One thing that he said that I asked him to clarify during a break was about aggregating. He said pre aggregating was the kiss of death. That is the exact opposite of what I believe so I asked him to confirm his statement. What he meant was aggregating and then throwing away the details. Some people in the industry say that you should aggregate your data and there is no reason to keep the details. I would agree with Ralph that that would be the kiss of death. In my experience, users ALWAYS want the ability to drill down to detail. How can you research things like data skew without details? What he confirmed to me is that pre-aggregating your details is a surefire time saver for querying and we also agreed that keeping the detail is the way to go.

He went on to cover some of the history of warehousing and the challenges and realities of the current environments. Of course he talked about his specific contributions, the data warehouse bus structure, conforming dimensions, and tying them back to simplicity and performance. He introduced the Bus Matrix which when tied to the data warehouse bus concept is very useful and simple to read. One of the last things he covered was centralization and the risks of being too centralized. Probably the biggest risk of being too centralized is the desire to do things in a big bang instead of incrementally.

The presentations by Informatica, Sybase and Trillium Software were mostly marketing. That's fair enough. They were paying for the day after all. I will speak about ETL tools in the near future. I got the chance to evaluate Informatica and two others some time back and I'll share my findings.

Sybase covered a concept I hadn't heard of and plan to research, columnar data storage. They call their implementation Sybase IQ. Instead of storing data in tables, they store everything in single column bitmaps. It reduces storage requirements and speeds access. It really seems fascinating. I'm still not sure of the details though so I want to spend some time researching this. One of the funniest moments came when the Informatica or Sybase rep, I forget which, said the columnar storage makes access infinitely faster. Ralph's engineering background kicked in and he broke in with the comment that "infinitely faster is a marketing phrase". The rep admitted that he should have said, noticeably faster. Ralph agreed that would be more appropriate. I thought of the comment, never bring an engineer to a sales call. Heh

And that got us to lunch. I had a turkey sandwich and then went back for a roast beef sandwich. We got to listen to Ralph and hear others ask questions. All in all a very good morning.

Well, that's the setup for my next blog. In that one I will talk about the afternoon sessions, which covered dimensional modeling and ETL. Instead of just talking about the seminar, I will get a bit more technical and define some of the attributes of a warehouse as well as share some of what Ralph talked about. In the very near future, I will actually walk through designing a mini-mart using a different set of examples than Ralph did but building on what I will talk about in the next blog.

A Day with Ralph Kimball, Part 2

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/003641.asp

Posted 3/28/2005

In addition to covering what Ralph covered in the seminar, I was planning on going into a lot of detail on designing a star schema from my point of view. However, after going over my notes I realized that would require a lot more "Days with Ralph". Instead, today I will cover what the topics Ralph covered, and then in the future write several star schema design entries. I'll show how to create a star schema based on a real life example that I have had.

So, on to the seminar! Please remember that this isn't what Ralph said as much as it's my interpretation of what Ralph said. I'm trying to explain what you'll get from his seminar but it's through my eyes not yours. That's why this is not a replacement for his seminar. Hopefully, I will induce you to attend if he makes it to your town. You should go just so that you can tell me where I got it wrong if for nothing else.

When we last saw our intrepid heroes they were devouring their lunch and asking Ralph inane questions. After the lunch break, Ralph covered Star Schema design and ETL. I felt the ETL portion was a little light but as all of this was covered in a single day, with some of the time consumed by vendors, I think Ralph did a good job. I'm not going to lay this out in the same order he did. I'm going to do this more in a topic outline fashion. I think this will flow better in a blog.

1. Keys

A primary Key (PK) uniquely identifies a single row in a table.

A foreign key (FK) is a link to a different table, i.e. a child or parent record.

A surrogate key is a sequential number that is meaningless. In oracle it's usually created by a sequence.

A natural key is a key that has meaning. An employee's social security number might be used to uniquely identify employees in an HR system (I would NOT recommend this but I have seen it). That would be a natural key.

Warehouse systems MUST use surrogate keys. It might seem like a pain but it removes administrative overhead when it comes time to fix production errors in the data.

A note is needed here though. In an OLTP system, an ORDER_ID may be a surrogate key. By the time that ORDER_ID gets to the warehouse, it is a natural key. It has meaning in the OLTP system. Warehouses need to take control of keys or they will fail. Do not use surrogate keys from an OLTP system as the keys in a warehouse. It sounds redundant but create a new sequence that will have no meaning to the OLTP systems. Remove that dependency.

Kimball recommends key fields be 4 bytes in length. This is primarily to conserve space. When a table can get into the billions of records, a few bytes per records really adds up.

2. Fact Tables

The fact table is a table that holds facts, or measurements. An event happens and a measurement is taken. When a measurement is taken, a fact record is created.

Facts are uniquely identified by a set of FKs to a set of dimension tables. The PK of the table may be all of the FKs but is usually a sub set.

This key implements the grain of the fact table. You can think of the grain as the frequency of the data, although that is somewhat inaccurate and very simplistic. A grain can be daily or monthly, order detail or order header, etc.

Fact tables can be aggregated into rolled up facts but the base fact table should be detail records at the declared grain.

Besides the PK, the fact table should contain numerical measurements. Sales quantity, sales amount, total profits, etc are all measurements.

In special cases the fact table may contain information other than FKs and measurements. A fact may contain a degenerate dimension. An example of this might be an order number. A dimension would be almost one-to-one with the fact table and in a warehouse this data would usually be display only. Instead of creating a monster dimension, you can carry this information in the fact table. Other examples might be a check number in a retail POS transaction or a risk score in financial transactions.

Indexing fact tables is a difficult proposition. If you only have one or two dimensions you can easily index them. What if you have 10 or 20? In this case you'll probably want a bitmap index on each FK. The actual facts in the table are almost never indexed.

To build a fact table you need to figure out where you're getting the data from, what's your grain, what fields describe the fact and what exactly is the fact.

3. The three grains of a fact table

Above I said that you could consider the grain the frequency of the data. In addition to that, there are three types of grain: Transaction, Periodic Snapshots and Accumulating Snapshots.

a. Transactions - This is the basic detail record of the warehouse. A POS transaction, an order details line, or a web "click". Kimball points out, and my experience aggress that this grain type is usually has a lot of dimensions and only exists if the detail event occurs. If the data from production is correct, there is no reason to change it.

b. Periodic Snapshots - This is a set of detail records that are repeated over time. I believe the example Kimball used was temperature in a room. If a system polls the temperature every fifteen minutes, the measurement would be the temp and the only dimensional change would be time.

c. Accumulating Snapshot - This is a fact that will change over time. The common example is the student enrollment example. The student's data doesn't change but the dates for enrollment, admission, etc.

In my opinion, type c is useful only when all possible date changes are known in advance and there are only a set number of possible dates. I have run into examples where this is not feasible due to the change, i.e. state, is dependant on other factors in the system. A transaction may have 1 state change and the life span of a single day or it may have 10 or 15 state changes over a lifetime of months. This is one of the things I will cover in my future star schema entries.

I think that covers the gist of fact tables. Ralph covered a bit more than that but he had all day. Heh

4. Dimensions

A dimension describes a fact. Dimensional data is usually textual and descriptive: the date of a transaction, the product type, the location, etc.

A fact record should connect to only one dimension record in a particular dimension.

Dimensions are independent of other dimensions. If you want to find data in a dimension using criteria in a different dimension, you must join through the fact table.

As I said above (well, really as Ralph said, I'm just regurgitating), a warehouse uses surrogate keys. The PK of dimensions should be a sequence. The one exception Ralph lists is for the date dimensions. He says that the key should be the physical date. This is doable in this case because the date is independent of application data. The date dimension should be a chronological table of dates with no gaps. It's highly doubtful that the dates will change.

Something Ralph said that resonated with me is that the dimensions implement the User Interface (UI). That is very true. If you think of a tool like BRIO or BO, the users are defining their interest using dimensions. It's very important to name the dimensions with meaningful descriptions of what is in them.

Dimensions should implement a NULL or Unknown record. The fact FKs should never be null. If the descriptive data is unknown, the FK should point to the Unknown dimension record. You can implement this with a -1 or with another entry from your sequence but the fact is, pun intended, that a fact should not have nulls in FKs. Since a fact is made of FKs and measurements, I think it is safe to say a fact table should never contain nullable columns.

A conformed dimension is a just a dimension used by two marts that match either in the dimension or in the values in the dimension. Basically it means making sure your definitions conform against different marts. This is also called Master Data Management and is the basis of Kimball's Bus architecture.

5. Changing Dimensions

There are three ways to handle changes to a dimension table: Over-write, Create New, or Save Old values.

a. Type 1 - Overwrite: This type is appropriate to correct errors. No history of the change is kept. You will probably have to recompute all aggregates that use this dimension.

b. Type 2 - Create New: This type is the most common in my experience. When you make a change, you create a new record with a new surrogate key and a new date. All future records will use this new record. When selecting facts, you can see the history of the values. Aggregates should not be affected.

c. Type 3 - Save Old Values: I have never had a use for this type. You would add previous_value columns and when there is a change move the old data to these columns and replace the normal columns with new data.

Ralph also talked about a Hybrid Type 2/3. His example was based on yearly changes to sales districts. I'll let you attend the seminar or his class to find out more. I have not run into this particular type.

Another topic he covered briefly was monster dimensions. If you have a dimension, the example was customer dimension, that is huge and has some often changing data, it is appropriate to split that dimension into two dimensions, one for static data, i.e. name, address, date of birth, and a different dimension for volatile data, income, number of children, etc.

Something else he covered that I cannot find in my notes is junk dimensions. If you have a set of flags that are always set one way or another, instead of creating a bunch of yes/no dimensions, create a junk dimension containing all of the flags. The final data set would be a Cartesian product of all the individual flags. You can also do this for other unrelated data that would make sense to combine. If all of the possible values result in a fairly small data set, say up to 10,000 records, you should probably pre-load the data. If it's more than that, you can populate it at load time. If a junk dimension might hit 1 million records, I would recommend splitting it apart. That's just too big for a junk dimension.

6. ETL - Delivering the data

Like I said above, the ETL portion was a pretty quick run through. We were running out of time. It's also hard to talk about specifics without talking about a specific tool.

Ralph started out defining E, T and L and adds M and R.

a. E - Getting the data into the warehouse
b. T - Clean and Conform
c. Prepare for presentation
d. M - Manage all the processes
e. Optionally R - Adapt to real-time

The basic ideas here are that E is for figuring out where to go for the data and how to extract it: Profiling, Change Data Capture or Extracts. T is data quality, i.e. cleaning, mapping to conformed dimensions and error handling. L generates keys, makes changes to SCDs, populates your fact tables, handles any late arriving data and does any aggregation. M is your workflow, job scheduling, backups, etc. The optional R is to handle loading in real-time. As data volumes increase and batch windows decrease, it's up to the warehouse professionals to figure out how to stream the data into the warehouse instead of batching it in.

Wow! I still have several pages of notes but I think that's enough for today. I am working on several documents on building star schemas, starting with getting requirements on through design, development and implementation. I'll put those out individually over the next couple of months with entries on unrelated topics.

Ill say this again. If you can attend this seminar and you're involved in your data warehouse at all, I would recommend you attend. You'll get some fairly interesting marketing literature from informatica, Sybase and trillium but more importantly, you'll get to hear this stuff from the guy who invented it. Having the chance to grill him on topics near and dear to your heart is fun too.

I guess I should also say I am in no way associated with Kimball, the Kimball group, Sybase, informatica or trillium. I'm just a geek who enjoys these kinds of seminars and I never seem to get to enough of them.

Ralph Kimball - Real-time Data Warehouse Design Challenges

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/006359.asp

Posted 11/1/2005

I "attend" webinars on a somewhat frequent basis. I can usually catch them around lunch time (I'm on the east coast), and since I don't have to travel it sometimes works out better than a live, in person seminar.

Of course, sometimes, they're complete crap. Or they are advertising and nothing else. They all have a bit of marketing, though. That's what pays for it, I guess.

Last week I attended one put on by Sybase and Informatica. It was a 1-hour, real time BI discussion; the title was Real-time Business Intelligence, Evolution in Information Access. What particularly attracted me to it is that Ralph Kimball was the keynote speaker. Remember that this is a 1-hour webinar so when I say things like "attended" and "keynote", I mean that in its best cybernetic meaning.

As you can probably tell from some of my very early posts, A day with Ralph Kimball, Part1 and A day with Ralph Kimball, Part 2, I like hearing Ralph talk. His topics are interesting to me and his obvious experience and skills are interesting to me. He's also a good speaker.

For this webinar, Ralph spoke for about 30 minutes. When he was finished, David Lyle, VP of Product Strategy for Informatica, spoke about Real-time Data Integration for Real-time Business Intelligence for about 15 minutes and Kapi Attawar, a Sr. Director at Sybase, followed that up for a 10-minute overview of Business Intelligence on Demand.

Something else very interesting to me was that the entire presentation was hosted by Stewart Cheifet. He's the former host of PBS Computer Chronicles. I always liked that show. The ad for the webinar also listed him as a moderator and broadcast journalist. He has a cool job.

If you are a Warehouse, BI or ETL guru, you will probably be bored by the presentation. Still worth the first half hour at least to hear Ralph, though. However, if you aren't a guru, this is an excellent 35000-foot overview of real-time warehouse and BI topics. There were a few terms new to me. Micro Batch ETL being one of them. I wonder when OWB will support that, eh?

Ralph used a phrase in this webinar that he also used in the seminar I attended earlier in the year, **Real-time is anything that is too fast for your current ETL**. I like that.

Some of the topics he spoke about were EII, CTF (I like this concept), SOA (THE buzzword of the day), SETL (this is what I use in my current project), and MBETL.

He also talks about a step dimension. This is another new one to me. A step dimension is used to model a sequential behavior. What he says is that "Most queries today look at flow at a single point" and we "Need to look at voyages through many points". He shows a sample schema with a step dimension. I helped out on a complex set of data marts earlier this year and a step dimension may have been perfect for that model.

He also talks briefly about compliance in a warehouse.

And after all this, Informatica and Sybase make their spiels. So anyway, if you would like to see a PDF of the webinar or see a replay, you can go to http://www.sybase.com/realtimebi. It's really a good show.

It only takes an hour and while it won't validate your existence as a human being, it might give you some ideas that you can research further.

ODTUG Bound! New Orleans or Bust.

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004558.asp

Posted 6/10/2005

Next week I'll be at ODTUG in New Orleans, LA. ODTUG is the Oracle Developer Tools User Group. What I call the ultimate Oracle techie conference. Lot's of speakers and lot's of good info. Thomas Kyte will be speaking in a special general session. I hope I get a chance to speak with him after.

This will be a great chance for me to network and maybe meet some of you, as well as learn some new stuff from some of the gurus. This conference covers data warehousing, security, coding, design, XML, Java, eveything. If it's an Oracle topic, it will probably be presented.

The conference runs from Saturday, June 18 through Wednesday, June 22. I won't be there Saturday. I'll be at a family reunion. My wife, son and I are driving together. Ten hours in a car with a two year old. I hope we arrive sane. We're splitting it over two days with plenty of stops. I'm from New Orleans so it's a great conference/vacation for me.

Anyway, I signed on to be an ambassador for some of the presentations. I guess I'll introduce the presenter and run get help if something bad happens. I get a special hat. That's kind of cool. All of my coworkers say that just confirms that I'm a real geek. I'm ok with that.

Some of the must see sessions for me will be "Ready, Set, XML" (John King), "Methods of Improving Data Extraction" (Ken Anderson), "Taming the VLDBeast" (Leslie Tierstein), "Real-time ETL" (Jean-Pierre Dicks), "Data Modeling on Steroids" (Bert Scalzo), and "Successful Dimensional Modeling of very Large Data Warehouses" (Bert Scalzo). I will be attending others but those are the ones I definitely want to attend.

If any of you are there, say hello. I'm the geeky, overweight guy with glasses. heh That just slightly reduces the candidate list. I also have a mustache and goatee. And a hat. An ambassador hat!

I'll try to post updates on the trip like James Koopman did in his entries about IDUG. It depends on what kind of bandwidth connections I can get. I may just write them daily but not post until I get back.

The Road to ODTUG

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004685.asp

Posted 6/20/2005

Getting to ODTUG, the Oracle Developer Tools User Group, conference took two days in a mini-van. They were the worst of times. They were the best of times. I traveled with my wife and two year old son. I knew we would be making many stops along the way to give my son, and us, a break.

We left Palm Harbor about 9am Friday morning. The weather was good. We got on Highway 19 heading north towards I-10. Gas was running about $2.09/gallon. This is a trip we've made many times. We go to New Orleans about once a year and we've driven to San Antonio a few times. This trip would be different because the family reunion we were attending on Saturday was in Florence, Miss, which is very near to Jackson and off the beaten path.

The plan was to stop over night in Mobile AL. We planned to take I-10 to Mobile, stop for the night and then cut across to highway 98 in the morning.

We ended upon spending a few nightmarish hours in Mobile. We detoured around town and then finally escaped. I like to call it, "the detour to hell".

We got off at Highway 90 at about 6PM because the map showed that road running into highway 98. It may do that somewhere but we never found it. As soon as we got on 90, we got the worst storm of the entire trip so far. It got dark like it was nighttime and the rain was so hard the roads immediately started flooding. Visibility was maybe 20 feet.

We must have missed a turn somewhere and ended up at I-165 and Wilson Ave. I would generously call this the bad part of town. Actually, I would call this the post-apocalyptic part of town. If you took East LA and dropped an atomic bomb on it, the survivors wouldn't want to live at I-165 and Wilson Ave.

Remember the guys from the road warrior? The killer barbarians with the spiked hair and leather? These guys would roll up the windows and lock the doors.

The stores on this road, the few that were open, were protected behind steel bars. The windows weren't quite so lucky. There were actually open stores with broken windows. I think there may have been a riot at some point. Or soon would be. I couldn't be sure. The rain was not helping my view.

Things did not get better when the power went out. All of the power for miles and miles. No more streetlights. The roads were flooding and there was no power. It was a neighborhood I didn't want to slow down in much less stop.

But things soon got better. My "Hey stupid, you're almost out of gas" light came on. Woowee. Doesn't get any better than that. Night falling, power out, in the rain and flooding, in a post-apocalyptic ghetto. And I'm gonna run out of gas. I have to admit, I was chagrined.

I swear we drove for a couple of hours and never left the bad part of town. For a little city, Mobile sure has a lot of bad part. I'd say for the most part it's a bad part. Scary bad.

We finally found our way somewhat and ended up on Springhill road. That led us to near a university and finally Highway 98 (back on plan!). Along the way, we drove down a road lined with huge oak trees and old Victorian houses. It quit raining and the power was on. Nice. I also found a very nice, well-lit gas station and filled up. I paid $2.19/gallon.

We decided not stay any where near Mobile for the night. We headed up 98 towards Hattiesburg, MS. It was getting late and we were getting tired and hungry. We pulled off 98 onto 198 at Lucedale MS.

There are no restaurants in Lucedale. Fast food but no sit down and eat kind of places. At least not that we found. And we drove for a half hour looking. I finally asked a Dominoes driver and he pointed us in the right direction. We never did find the promised restaurant but we did find a subway and a hotel. The subs were scrumptious. We were starving.

The hotel billed itself as the newest hotel in town. I paid $48.15 for a room with two double beds for the night. Not a bad rate.

It didn't occur to me to question a permanent sign that says a hotel is the newest in town. It might have been the first hotel in town. I think it was also the only hotel in town. I got a non-smoking room that smelled like smoke. Could be worse. Could have had bloodstains on the mattress. Speaking of which…

As I pulled the blanket out of it's tuck, it uncovered the mattress. I swear it had old bloodstains in a splatter pattern. I checked the other mattress and it had what looked like old bloody leg imprints. I didn't uncover the entire mattress. I didn't really want to see what came after the bloody legs. Ick.

But we did sleep there. And that ended my Friday.

Saturday was a much better day. We left Lucedale, MS after filling up with gas, $1.99/gallon, and had breakfast at Shoney's in Hattiesburg. The food was good as it always is at Shoney's.

We had a good drive to Florence and I got to hang out with relatives I haven't seen in years. I hadn't seen some of them in 20 years. And we ate a lot. Fried chicken, casseroles, dressing. Good stuff.

The area around Jackson is beautiful. I don't think I'll ever live in Mobile but I could live in Jackson.

I saw gas as cheap as $1.91/gallon.

One note I wanted to make. I made a trip to Best Buy on Wednesday night and bought a mobile DVD player. We strapped that sucker to the seat and it paid for itself in the first hour. You wouldn't believe how big my son's eyes got when he realized he could watch JoJo while we were driving.

And now it's Sunday. I'm writing this at my Mother's kitchen table. We're staying here for the week. I'll be heading to ODTUG in a while. I'm hoping there is broadband access from the hotel where the conference is being held. I hate dial up. If I can't get broadband, I will post this using my dial up VPN but I'm hoping for a faster connection.

At a minimum I plan to attend the Ambassador meeting and Thomas Kyte's presentation. If I get there early enough, I'll go to some of the sessions.

ODTUG - Sunday: Day One

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004694.asp

Posted 6/21/2005

I guess I should have called this My Day One. The first day of the conference was Saturday but I was at a family reunion instead. So that would make Sunday, June 20th, my first day at ODTUG NOW! 2005.

Hopefully I've made it plain that the ODTUG conferences are the Oracle techie conferences. I'm a member of ODTUG but for some reason, I didn't get a member sticker on my badge. I'll have to check into that.

ODTUG is the Oracle Developer Tools User Group. Historically, that means a conference about the tools Oracle provides for application development. The charter has recently changed though to include any tools that are used to develop on Oracle, even third party. I'm not sure if that's a good thing or not. I guess time will tell.

Anyway, on Sunday I got a late start. We left my son with my mom. My wife and I walked around the French quarter for a while and then had lunch at the Desire Oyster Bar across the street from Old Absinthe House. The oyster bar has a copper ceiling. There are designs pressed into it and it's screwed into the ceiling. Very cool and very New Orleans. We both got red beans and rice with sausage and shared a crawfish corn cake. Mmmmmm. Can you tell where my priorities are? ;-)

Anyway, I eventually got her back on the road and I went to the Sheraton New Orleans to register. The Sheraton is on Canal St which borders downtown and the French quarter; just a couple of blocks from Bourbon St.

The registration was easy. Everything was prepaid so I just had to pick up my bag and other assorted goodies and then walk around. The conference comes with a Dulcian polo shirt. They were out of x-large. I won't try to interpret what that means about oracle developers. heh

Quest is one of the vendors (I'll make a note to myself to list all of the vendors in a future entry). I filled out a slip of paper with my contact info and got another t-shirt. He gave me an x-large without asking me. I won't try to interpret that either. ;-)

I attended the Ready, Set XML session by John King. John has his own training company, www.kingtraining.com.

I'm very into XML. I've been using it since 1999 and have found all kinds of uses for it. For some reason it's still a pretty big unknown to most of the Oracle developers I talk to. I'm not sure why. It's definitely not sexy and it's usually anti-relational but it makes living in a heterogeneous world a lot easier.

Anyway, it was a good presentation. John is a good speaker. He had one slide that should have shown up red and instead showed up hot pink. His comment about that being appropriate in New Orleans kind of fell flat but most of his jokes got a laugh or two.

His handouts didn't match his presentation completely which is kind of a peeve for me at conferences but he went into more detail than the handouts went so I guess that's a good thing. He promised to get the new slides up on the ODTUG site as soon as possible.

He covered the differences in structured and unstructured storage in detail (CLOB vs. Relational), which is good. I don't think most people understand the differences there.

He covered two topics that made me glad I attended.

There is a new feature called copy evolve. This is to evolve existing stored XML when an XML schema changes. I didn't even now this existed. He didn't seem impressed by it and went over the limitations but I plan to hit the docs and get a better understanding of this feature. If you use XML much in the DB, you'll understand how important this COULD be.

The other interesting topic was Oracle's implementation of WebDAV. I knew this was there but just haven't had the need to use it in the real world. WebDAV lets you access the DB and data using MS Explorer (Not internet explorer, just the basic file explorer), a browser (including firefox which he used) and even office tools like Word and Excel.

He used the WebDAV demo that's on OTN. No new info really but he did walk through installing and talked about some gotchas.

Because this is a topic I'm very familiar with there were a couple of small nits that I could pick but all in all, I'm glad I made it.

My next session was the Ambassador meeting. An ambassador introduces speakers and fetches help if there are any problems during the presentation (AV equipment problems. No one is anticipating riots or anything). I got my ambassador hat. It's a great big gaudy Mardi Gras hat. I'm not promising to wear it in public.

I have to say I'm a little nervous introducing people. I haven't done anything like that before. But I'm hoping that by next year I will be presenting so it's a good place to start.

The sessions I'll be ambassador for are: Methods for improving data extraction and Code generation in the world of Business Rules on Monday, Oracle Application Server 10g: Best application server for the Oracle database on Tuesday and Migrating from MS-Access to HTML DB on Wednesday.

The last session I attended for the night was a special general session by Thomas Kyte. I think everyone who reads this blog knows who Tom is. If you have never seen Tom present, you need to make the effort to do so. The material was great and he is very dynamic. And he is also incredibly funny.

His slides about what he's learned from TV had me rolling. Things like all cars burst into flames on impact and foreign people really prefer to speak English between themselves. You have to see it.

His presentation made two important points. Primarily it said you have to keep up but it also said you have to question authority.

By keeping up, that means keeping up with changes in the database (or OS, or Hardware, or whatever else). Read the concepts manual with each release. What might have been true in v7 might not be in 10g. Probably won't be in 10g. Performance tuning is particularly version dependant. I think he mentioned that there was something like 350 new pieces of functionality between 9iR2 and 10gR1. You'll only get that from the new features guide and the concepts manual.

The topic of question authority also applies to more than just the DB. Anyone can call himself or herself an expert. If you can't ask them why they say something is true or false, I would immediately mistrust that expert. If they present no proof, I would mistrust that expert. Always ask why. Always expect proof.

I can tell Tom gets a lot of practice speaking. He seems very comfortable doing it. I think the fact that he knows his subject so well helps. Tom also doesn't just regurgitate his overheads. The slides give you the basics of what he's talking about but he really says a lot more than you would expect just from reading the slides.

I hope that when I finally do present, I will be nearly as comfortable with it. I think I'll start with a smaller crowd though. Just about everybody at ODTUG who was there by Sunday showed up for Tom's talk.

I introduced myself afterwards and we spoke for a few minutes. He's very personable. A few people hung out and asked some specific questions. He seemed glad to help them. I would have offered to buy him a beer but my wife was already on the way to pick me up. I'll just promise myself to buy him one in the future.

Speaking of that I guess that's one downside of mixing a conference with a vacation. The night time events are the best time to network but given the choice between networking or dinner with my wife, I choose my wife. If I was there by myself, I would attend all of the functions.

So I jumped in the mini-van at Magazine and Common and went back to Mom's for half a muffalotta and some Zapps crawtaters for dinner. Speaking of Zapps, they will send Zapps spices to the military for free. Check it out.

ODTUG - Day Two

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004718.asp

Posted 6/22/2005

Monday was a bit more disappointing than Sunday. The sessions I went to covered XML, Business Rules with code generation and data loading and extraction. The topics were interesting but not quite as much as Sunday or as much as I expected. The topics did get me thinking though.

I wrote earlier about my day last Thursday and my prototype data load. The data loading topics at the session just reiterated how hard it is to get data into a system. If you are loading straight data into flat tables it's easy but add in a few foreign keys or unique indexes and all bets are off. I had hoped to get some insight into alternatives to the way I have been doing it from the data extract/load presentation but didn't get that.

As an ambassador, I have to introduce the speakers and give a couple of directions. I first ask them to turn off phones and pagers and to please fill out the session evaluation form. That form is used to select the best presentation, best speaker, etc. I have to admit I didn't wear my hat on Monday.

I introduced two speakers on Monday. I can't believe how nervous I was. When I introduced Ken Anderson for the data extract and loading presentation, I think I may have introduced him in French. I can't really remember. Of course, I don't actually speak French so I'm not sure how that worked out. I didn't hear any phones ringing so I guess people understood what I was asking them. ;-)

Ken is a good presenter. His presentation was based on 9i and some of the points he made are no longer as important with 10g. He did engage the audience though and I think everyone felt it was a good discussion. He had it categorized as Intermediate and I think that is dead on.

After that, I went to my next ambassador gig, a presentation on code generation using business rules by Michael Rosenblum of Dulcian. Michael had introduced himself to me the night before at the ambassador meeting.

When I got to the room, one of his co-workers was handing out the handouts and session evaluation forms. After speaking with her for a moment, I was under the impression that she was going to be the ambassador for the presentation. That was fine with me; two in a row is a bit much. Unfortunately, just before time to begin, she came by and said I would have to do the introduction and ducked out. Now that is wrong! I had to run up to the front and read the Guy's bio real quick. When I introduced him, I think I said something like, "Grzpt blogu ma finga dym zraba". Then I pointed at him and sat down quickly.

This one got me thinking about my prototype again. I actually wrote a code generator for that project. Basically, once a day I have to generate a couple of hundred aggregates across 50 million or so logical groups against 1 billion details records across 10 time dimensions. There are many ways to do that: truncate insert, merge, CTAS, etc. I wrote a code generator so that instead of re-writing the various queries when I want to try something different, I can just change a few parameters and regenerate. Basically, Michael spoke about the same concept but in a production environment instead of a prototype one.

He also covered the differences between interpreting and compiling generators. With an interpreted environment, you are more dynamic but tend to take a performance hit. With a compiled environment, it's faster but you have compilation dependencies between modules. My particular implementation is compiled.

This presentation was categorized as Intermediate. I think Novice or All might have been more appropriate but it wasn't a huge difference. Michael's boss Dr Paul Dorsey was in the back making occasional comments about the presentation and/or about Michael's delivery and I know that threw him off a couple of times. I think giving him some critical feedback after the presentation would have been a better method.

I spent some time taking a break and staring out of the window from the third floor down on Camp St watching people walk by. Very relaxing. I also checked out the smaller buildings near by. The Sheraton is huge. A lot of the smaller buildings have moss and/or grass growing in random spots on the sides. I'd never sent hat before. There is even one building just across the street that has a small tree growing out of the edge of the building. That can't be good for the brick.

The afternoon sessions were nothing to write home about. I did get to meet up with some co-workers from the past. In another life I was an Oracle Applications person (functional/technical for HR and Payroll). I met up with some people from Yale where I worked implementing their HR/Payroll and a friend of mine I've known for over 10 years (and worked with at several places including Yale). This is another perk of conferences, I guess.

At lunch we had a couple of speakers talk about Jautomator. From what I can tell, it looks like forms, runs like forms and smells like forms. I'm not sure why I wouldn't just use forms. Unless, maybe Oracle, despite denials, is not going to continue supporting forms. The phrase, he dost protest too much pops into my mind. Duncan Mills is doing an awful lot of Java presentations this year.

Speaking of sales spiels, I still have some misgivings about the change in direction for ODTUG. ODTUG right now is for techies by techies. I have been to a lot of conferences and ODTUG is by far the most technical oriented. I am not opposed to offering technical presentations by techies on third party tools. I am opposed to vendor presentations. But, as I think I said in my last entry, only time will tell.

I got my member sticker on my badge finally. I just went and asked. I think when my company filled out the registration they didn't click the member button. Which also means they didn't pay the member price. I did tell them I was a member though.

I have a digital camera but didn't bring the cord for it. I'll post a picture of my ambassador hat when I get back home to Florida.

For lunch we had some chicken and black eyed pea concoction the hotel made up. It was good. For dinner I had Mom's red beans and rice. It's a Monday thing.

That's it for today. I hope everyone is enjoying these posts.

LewisC

Comments

Bruce Bergman 6/22/2005

Enjoying your blog. :-)

One specific comment: our change of direction (or, maybe, our clarification of purpose) absolutely does not have room for increased vendor presentations. While we are now allowing other, third-party tools into the mix, we are not soliciting -- or condoning -- any kind of vendor-slant to presentations. There will always be vendor presentations at our conferences, but just as they are today, they will continue to be clearly identified as such.

So, the obvious question is: how do we plan to allow more presentations on other tools without making them vendor presentations? Good question. :-) First off, if the person giving the presentation is from the company who makes that product, then that's a good indicator that the paper will be rejected (or at least sent to the vendor presentation category). If the paper discusses techniques, methodologies, processes, ideas, etc., then that's a good paper. If it compares approaches or tools, then that's a good paper. If it illustrates a unique solution or approach to solving a problem that involves Oracle, but uses a non-Oracle tool, then that's a good paper.

Trust me, the board did NOT make this decision in order to bring in more money or increase revenue. It was honestly and strictly a result of what the membership is telling us. And they're telling us that they no longer use just one tool, or Oracle-only tools. They use a variety of tools in the job, and they would get more "bang for the buck" at our conference if we educated them on those other tools. For example, David Scheliss was one of the presenters who were nominated for Best Presentation. You know what his presentation was? Using PHP with Oracle. That's a perfect example of the kind of paper we want.

I hope this helps alleviate some of your concerns.

Thanks!

Bruce A. Bergman
ODTUG Vice-President, 2005-2006 Conference Chair

ODTUG - Day Three

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004733.asp

Posted 6/23/2005

Tuesday turned out to be a lot more interesting for me. I got a couple of Vendor demos (at least one of which I plan to review in the future) and attended a couple of superior sessions.

I only acted once on Tuesday as an ambassador for "Why Oracle 10gAS is the best app server for Oracle" by Tugdall Grall of Oracle Corporation. Personally, I felt it was informative. I may have been in the minority. We started out with 26 attendees and about 10 walked out at various points midway. Yes, I wore my hat.

He talked about a LOT of topics. One thing he spoke about was the ability to debug Java and PL/SQL in the same session using JDeveloper. I didn't realize you could do that. The last time I tried (two or three years ago), I could either debug the PL/SQL or debug the Java but not both. That has a very high cool factor. He also spoke about ASLM Diagnostics, which is something I will have to research. I don't know enough to comment at this point but it peaked my curiosity. This session was categorized as All. That is appropriate for it I think.

My next session was a presentation on bulk collect and bulk binding. This particular session was categorized as Advanced. I use these features fairly heavily and really wanted to get some new information. Unfortunately, the presentation could have been created straight from the Oracle docs; possibly from the concepts manual. I would have categorized it as Novice. It was very technical, but very basic, knowledge.

After that was lunch, a good surf and turf with mashed potatoes.

After lunch I put together some thoughts about what I was seeing as successful and not so successful presentations. Some things I dislike are no handouts, too many topics in one session and regurgitating the slides rather than talking about the info on the slides. After talking to quite a few people, a lot of people feel the same.

I also got a lot of comments about not being able to understand the speaker. There were quite a few presenters with accents. I think several are Indian (I'm guessing here) and, oddly enough, several are French. A person can't control what their accent is so I'm not sure what to recommend about. I think everyone should have the opportunity to present. I'll try to get more opinions about that on the last day of the conference tomorrow. What do you think?

The sessions after lunch were amazing. I attended Designing Data Warehouse for High Performance by Vincent Chazoor. During the course of his presentation he brought up a couple of ideas that I disagree with but he really knows his topic. He obviously has done this in the real world and has done his background research. This is one of those presentations where reading through the hand out I would have liked to dislike the session but got sucked into it in spite of myself.

Right off the bat he said to quit arguing about Kimball vs. Inmon and get on with the designing. There are enough things to work on without spending all of the time arguing. How true that is. He covered the pros and cons on various relational vs. dimensional models and even how Oracle has supported joins, star joins and star transformations in different versions of the database.

One thing he said that I immediately disagreed with was putting the natural key of type I/type II combination SCDs in the fact table. That seems contrary to everything I've done in the past. I've been giving it a second think and it might work out in certain situations. Vincent provided a link that I will be reading but haven't yet.

And now I'll have to research the "Throw Away Method". This is a way to optimize joins based on a white paper by Martin Bell. I had never heard of this before.

He also covered several methods of handling hierarchies and ETL improvements like Partition Exchange, Materialized views and some ETL guidelines. Unfortunately we ran out of time. This is one of those situations where too many topics were covered in a single session. We probably didn't get to cover the last third of the presentation. If Vincent does this again next year, I would like to see him split it into two or maybe three sessions and get a little more specific. He's a good speaker and seems comfortable up in front. This session was categorized Intermediate. I think it hovered somewhere between Intermediate and Advanced.

My final session of the day was Bert Scalzo, of Quest Software, on data modeling on steroids: patterns and reusability. I don't know if Bert is a natural speaker or just has had a lot of practice but he's good at it. He genuinely seems to enjoy being up there; he paces it well and has a self-deprecating sense of humor. He also knows his topic in depth.

I don't judge people by the degrees or the certifications they have. To me it's all about experience and the desire to excel and learn. But I have got to mention some of Bert's credentials: BS, MS and PhD in Computer Science, he also has an MBA and holds several Oracle Master certifications. He has insurance industry designations FLMI and ACS. He's written two books and many articles. He's worked for Oracle as a trainer and as a consultant and he's been a DBA for Oracle versions 4 through 10g.

Of all of his credentials I would say the last line above is the most important. For the degrees and certifications, I would have to say he's either really smart or has a lot of free time. ;-) I'd guess the companies he's worked for must be a lot more willing to pay for certifications than the companies I've worked for.

The presentation was based on a book by David Hay. He didn't give the name of the book but he said it was small and brown and is available on Amazon. The first several slides were advertisements for Quest but he blew through those pretty quickly.

What he really covered was what are patterns and how do you approach reusability. Rather than reinvent the wheel, spend time identifying what patterns are available. A customer is a customer. You may have sub-types of the basic customer for specific lines of business but they are probably pretty close.

He went into Intersection Patterns and covered Roles and Assemblies. Roles can greatly reduce data model size and assemblies enable flexible database design. Assemblies are also called BOM solutions. The drawback to both roles and assemblies are additional time and complexity coding against them.

This presentation didn't really break any new ground for me. I use roles pretty regularly and have used assemblies a few times. I didn't know they had a name. I just knew they were the best I could come up with for specific tasks at specific times. It was one of those, "Wow! I'm smarter than I look" moments.

For dinner it was Don's Seafood Hut and desert at Gambino's. Can't beat that with a stick!

I would have to say Tuesday was the best day so far.

For a sneak peak of day 4, I ended the day with sessions by Don Burleson and Mike Ault. I'll be leaving early Friday to return to Florida so I won't be able to post Day 4 until Monday.

Thanks,

Lewis

Comments

Thomas Kyte 6/23/2005

... I also got a lot of comments about not being able to understand the speaker. There were quite a few presenters with accents. I think several are Indian (I'm guessing here) and, oddly enough, several are French. A person can't control what their accent is so I'm not sure what to recommend about. I think everyone should have the opportunity to present. I'll try to get more opinions about that on the last day of the conference tomorrow. What do you think?....

I think - when I speak in Europe and other places, I probably "have an accent". Heck, I have an entirely different language.

the goal, speak so that those who are listening can understand.

http://tkyte.blogspot.com/2005/05/precise.html

It is hard, and I'll be trying harder in the future myself.

I feel for those that are speaking to us whose first language is not english. I try to judge them by the content and the information.

And I feel bad when I travel to other places and cannot speak the language.

wish our education system would prioritize languages more, it really does count later in life and as a 9th grader, I did not realize this

Believe it or not, most people on the planet don't speak english :)

LewisC 6/27/2005

Tom,

I completely agree. I also think that there are few enough people doing these sessions that I would hate to limit the number of available presenters.

Thanks,

Lewis

Vincent Chazhoor 7/7/2005

Thanks for mentioning my presentation here. Please see the link to "throw away method" paper by Martin Berg http://www.miracleas.dk/tools/throwaway2.pdf

ODTUG - Day 4

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004772.asp

Posted 6/27/2005

Wednesday, the last day of the conference was a good one. For the most part, all of the sessions were worth the time and most were also very informative.

How it was setup

Before I talk about day 4, let me describe how the conference was laid out. I probably should have done this the first day but I didn't think about it.

At any particular time, there were 9 sessions running concurrently. Most session ran 45 minutes to 1 hour. In addition to these sessions, there was hands-on-training running in a side room. The training was an additional fee. The training sessions were accessed via two hallways, rooms a1 and b1-b3 in one hallway and c1-c3, d1 and d3 in the other. In front of the hallways were the vendors. Special sessions that required larger room, like Tom Kyte's on Sunday, were on a different floor.

The sessions ran from about 8:30 to 5 with evening and/or nighttime events on most nights. I didn't attend any evening events. That's about it. Pretty basic layout. I asked the conference management about how many attendees there were. They said 600 showed up. It seemed like a bigger crowd to me.

So many sessions, so little time.

On Wednesday, I got to learn a bit more about HTML DB. One of the sessions was given by Scott Spendolini, the HTML DB product manager. HTML DB is a new interface that comes built into the database. All versions: EE, Standard and Standard One, support it.

HTML DB is a RAD screen designer/code generator. The screens are built using meta data. The screens that define the screens were built using HTML DB.

Based on one of the HTML DB presentations I attended, HTML DB is a great tool for migrating away from MS-Access. If you combine HTML DB with Oracle Standard One, the price is reasonable to do so. The benefits that you immediately receive are security (you can download a password breaker for MS-Access), scalability, availability and oddly enough, portability. Comparing MS-Access to Oracle, the benefits of scalability and availability are obvious, but portability? With MS-Access, you are really stuck to one OS. With HTML DB, your data is available on any platform with a web browser.

HTML DB likes single column, numeric keys so some re-engineering might need to be done. You probably want to do that anyway as most MS-Access databases are not designed by data modelers. Scott used the migration workbench and walked step by step through the migration process and through building several screens. He did all of that in one hour and still had time for a HTML DB 2.0 demo (coming soon).

I've actually played with HTML DB somewhat at home but I wanted to see it in the hands of a pro. Scott has a lot of experience with this tool.

One thing I did notice though is a lack of knowledge transfer between product teams (not just here but in general). What I mean is that Oracle is doing so many things that the techies and product managers aren't really aware of what other groups are doing. One example is that in the MS-Access migration session, an attendee asked about using mail merge (a big benefit for some people to stick with MS-Access). Scott talked about a downloadable component that maps the RTF structure of WordPad and after some kludging would allow mail merge in Word. A tedious and error prone process at best.

However, as I learned from one of the XML DB sessions on Sunday, using WebDAV and XSLT, this would be a native and highly intuitive solution requiring no mapping to file formats. I mentioned this but I don't think Scott knew what I was talking about.

The sessions I attended after lunch were all about tuning. I managed to catch two sessions back to back by Don Burleson and Mike Ault.

The first, Oracle 10g SQL Tuning Secrets, was given by Don. Don has written many books on tuning and participates on several web sites.

At the beginning of the session, he spoke about RAMSAN and an SSD specific version of Oracle. I hadn't heard about that but it intrigues me.

Anyway, the gist of the session was things developers and DBAs can do to optimize the DB and/or the application. He spoke about some of the obvious things: adding needed indexes, use materialized views, cache small, frequently accessed tables, tune complex queries to reduce redundant access, etc. He also spoke in depth about parallel query and when to use it.

Two things the he proposes that I feel are arguable are using hints and temporary tables. He said developers should spend the time to get the best plan and then to "cement" that plan by using hints. I disagree with that. In a completely static system, that might work but I think it's asking for trouble in the real world. As far as temp tables go, I have seen them improve performance for some queries but I believe that had the queries been re-written appropriately, the same gain and possibly more could have been gotten without temp tables. I don't doubt that Don has seen more databases and application than I have, though so his experiences would be different from mine.

He also covered AWR in detail. The last section of his presentation was on AWR and table and index usage tracking over time. He ended his session with the blurb that "AWR builds a platform for AI-based database tuning." That's a pretty cool concept.

Over all I liked the presentation. Either Don has given this presentation many times or he practiced it quite a bit. He has it down perfect. The only downside was that there was so much information and he spoke so quickly, there was never time for questions. I would have liked to ask a question a couple of times but he never slowed down enough for me to get it in. Don also uses humor to good effect. He made the comment that TOAD has gotten popular because OEM sucks. ;-) That got a laugh from most everyone.

Don also spoke about an organization that trains miniature horses to be guides for blind people. It's not Oracle related but I thought it was worth a note. I didn't realize such a thing existed.

The last session was with Mike Ault. Mike has written MANY books on Oracle. His session was "Using Statspack to track down bad code".

He talked about the differences between STATSPACK and AWR. For one, AWR is an additional license and statspack is not. He also covered how to read a statspack report.

There's not a lot I can say about the technical aspects of this session. It was very detailed. He listed some things to look for when using statspack. He covered the usual: use bind variables, identify performance goals before tuning, spend time tuning the right things, etc.

Mike is another presenter very knowledgeable about his topic. We spent a lot of time in this session in a question and answer mode that I personally found very useful. One thing he came up with that I plan to implement in my code is to always include a comment in your code identifying the module it's running in, i.e.

```
SELECT /* my_pkg.proc1 */ SYSDATE FROM DUAL;
```

That way, when you get the SQL from V$SQL or STATSPACK, you can immediately tell where it's coming from and not have to dig through a lot of code to find it.

I would gladly go see Mike speak again.

Wow! Finally the last day of ODTUG. Talk about an exhausting week. Combining a conference and a vacation is not a good plan. I would have been better rested at work!

I will post one more entry about ODTUG after this one. That one will include some final thoughts, a list of vendors I spoke with and my bling. Conferences are always, all about the bling! After that, I will get back to my technical topics. I still need to post the final Sorcerer of OO (Object views and tables).

Thanks for reading this far. I hope you have enjoyed reading these as much as I enjoyed attending ODTUG. Next year, you should attend.

Comments

Thomas Kyte 6/27/2005
Just an FYI

Comments are stripped out of plsql (9204 or 5 and above, forget the patch release that did that) in order to generate more sharable SQL. dbms_application_info lets you tag the SQL:

```
ops$tkyte@ORA9IR2> alter system flush shared_pool;
System altered.

ops$tkyte@ORA9IR2> begin
2 for x in ( select /* my_pkg.proc1 */ sysdate a from dual tkyte )
3 loop
4 null;
5 end loop;
6 end;
7 /
PL/SQL procedure successfully completed.

ops$tkyte@ORA9IR2> begin
2 dbms_application_info.SET_MODULE( 'my_pkg', 'proc1' );
3 for x in ( select sysdate b from dual tkyte )
4 loop
5 null;
6 end loop;
7 end;
8 /
PL/SQL procedure successfully completed.

ops$tkyte@ORA9IR2> select action, module, sql_text
2 from v$sql
3 where upper(sql_text) like 'SELECT SYS%TKYTE%';

ACTION MODULE SQL_TEXT
-------- -------- ----------------------------------
proc1 my_pkg SELECT SYSDATE B FROM DUAL TKYTE
SQL*Plus SELECT SYSDATE A FROM DUAL TKYTE
```

the action and module stick with the SQL -- the comments are normalized away....

Scott Spendolini 6/27/2005
Lewis,

Thanks for the kind comments - I'm glad that you enjoyed my presentations.

Regarding the solution for Mail Merge - I agree that XML DB, WebDAV and XSLT is a more "solid" approach, but that involves some setup on the database & application server tiers. Using the RTF solution involves nothing more than being able to edit an RTF document in WordPad and upload the results into HTML DB.

Either way, I think that the core of this discussion highlights how flexible HTML DB can be when integrating these "one-off" type solutions. This also helps up as Product Managers determine what functionality needs to be merged into the product.

Thanks,

- Scott -

LewisC 6/27/2005

Tom,

Thanks for the update. I'm glad I didn't change any of my code. ;-)

I'm also already using app info in several places. It didn't occur to me to track it that way. I'll start using app

info more often. Actually, I have a monitor/debug package I call everywhere. I could add the call to app info

there. Hmmmm. Thanks again!

LewisC

LewisC 6/27/2005

Scott,

Thanks for the post. I hope my comment didn't appear derogatory. I didn't mean it that way. You guys are doing so much, I imagine it's pretty hard to keep up.

For the RTF, I thought you said it was written for wordpad and to use it in word would require some mapping of RTF codes? I may have mininterpreted that.

But you're right. HTML DB is very flexible. We are already planning a couple of in house apps where it will be a perfect fit.

I also meant to say that you are the person who wrote the HTML DB studio, http://htmldb.oracle.com/studio. I missed that in my notes somehow.

Thanks,

Lewis

Scott Spendolini 6/27/2005

Lewis,

Your comment didn't appear derogatory at all - no worries!

To clarify the RTF issue - when you use Word to save an RTF file, it adds a whole bunch of "stuff" that WordPad doesn't. Thus, it's far easier to parse through the source of a WordPad RTF file than a Word RTF file.

To see the difference: Open WordPad, enter "Hello World", save the file, and then open it in Notepad. Repeat for Word (saving as type = RTF).

In the interest of time, I chose to only address WordPad RTFs in the PL/SQL procedure which I wrote which parses out the file. Given more time, it's entirely possible to do the same for Word RTF files.

Either way, this solution allows the end user to create a document in WordPad (which IMHO has all the features one needs in a word processor) using #TOKENS# where field data will appear. Any font, size, color, etc. can be used, giving a basic non-technical end user great flexibility to create a report of almost any basic layout. Once the document is created, it can be uploaded via an HTML DB web page, and then reports can be run with it.

Thus, this solution is ideal for the basic user, as it uses tools already installed on their desktop - a web browser & WordPad. No additional software - on the client or server - is required.

Thanks,

- Scott -

LewisC 6/27/2005

Scott,

That makes a bit more sense to me now.

Congrats on being selected for the 2005 Oracle Contributor of the Year Award from ODTUG!
http://www.odtug.com/

That's great.

LewisC

Scott Spendolini 6/28/2005

Thanks, Lewis! It was quite an honor!

ODTUG - Final Thoughts

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/004842.asp

Posted 7/4/2005

I promise today will be the last time I write about ODTUG. I just want to share some final thoughts about the conference, a run down on some vendors I spoke to and my incredible collection of blingage.

Final Thoughts

For an Oracle geek, this is the conference to go to. I've been to others and until now, IOUG was my favorite. I've even been to conferences for other vendor's products and this one tops them in pure technical content. This is a conference I would like to attend every year.

I wrote earlier about some people having a problem understanding some of the speakers. I spoke to more people on the final day and a some people did think it was a problem. Not in general but just for a couple of sessions. I don't know if that is a minority or majority view but I would rather make an effort to understand someone and learn something than to not attend a session because someone has an accent. It does make it a little bit harder but, hopefully, the information you gain is worth it.

This conference covered a lot about business rules, meta data, java and warehousing. Oracle is really pushing java. I've read the statement of direction for Forms but from the way Oracle is pushing jDeveloper, I don't think the future is very bright for forms. This is a tool I would love to see Oracle open up and give to the open source community to enhance and maintain. The architecture is great.

Business rules and warehousing seem to be on everyone's mind. I know warehousing is on mine. Working with vast amounts of data requires a different viewpoint and a different approach. I'm also getting into code generation more and more. I expect to see these two topics back again next year.

Vendors

Some of the vendors I got to speak with were Quest, Kumaran, Confio, Maverick Solutions, TUSC, and JavOra Enterprise Solutions.

I plan to download and review Confio's product DBFlash for Oracle. This product is a performance monitoring tool based on waits. You can see where your waits are and drill down to the actual SQL causing the problem. Looks very interesting. I got a quickie demo there and want to download and do my own review of it. I'll post back here when I do.

Another vendor I spent quite a bit of time with is one you may not have heard of, JavOra. Right now JavOra doesn't have a commercially available product. They sell only to US DoD. The primary tool that they develop, that I got a demo for anyway, is a product called jEM. jEM is a dynamic menu system for managing J2EE, Oracle Forms and Portal applications. The thing that got me was that the same menu runs natively in forms and in J2EE applications. And there's no programming required. Just plug it in.

jEM provides security for your application also. Users can be validated through the DB or through LDAP or another web service. Access is based on "Users" and "Roles".

Seeing how the military is built around bases, jEM has great "Site" support, so that you can define a menu or a role that will only run at specific sites or that is shared amongst sites. There are ease of use features like identifying a menu as common and everyone automatically gets it.

I was really impressed with jEM. I can't do it justice in this little bit of space. It really needs pictures. Check out the web site, http://www.javora.net. A downside here is that you really need IE to view the interesting pages.

Bling

And now the bling. This is a picture of me with some of my bling! I know, I look really goofy. It was a long week.

The hat I'm wearing is the infamous Ambassador hat and yes, I actually wore it for several presentations that I introduced. I've also got my two favorite blings, the TUSC V$VIEW layout poster and the TOAD Handbook.

Some other goodies are some CDs with software from Quest, Oracle and ODTUG (presentations on that one). I got a keyboard layout for TOAD. It's little long for my keyboard though so it's sitting on my desk. I got the ODTUG backpack and that came with a really cool Rubix Pen. It's like a Rubix cube but you can write with it.

The hat is my favorite. It's sitting on the top of my cube at work next to the paper crown. You can see it from everywhere in the cube farm.

I also won a raffle and got an Ultra Mini 4-port USB Hub. I got it in the mail on Saturday. It's perfect for carrying with my laptop.

It was commented on how professional most of the presentations by Oracle employees were. I have to agree. One person told me that Oracle sends employees to presentation training if they are going to do presentations in a venue like ODTUG. I don't know if that's true but what a great idea. Every company should do that.

Comments

Phillipe D. 7/4/2005

Yes, the ODTUG attendence seemed much larger than 600. One vendor that stood out "of nowhere" were JAVORA. While they do not produce an automated tool for migrating Oracle Forms to J2EE they have evaluated all these products across the past 3 years under DoD contract. Recommend checking with them before picking a tool, they are really "in the know" with the tools and the vendors.

What really put JAVORA over the top for me was jEM. Besides the features you mentioned the speed was incredible! On WED they showed an emulator module that allows an adminstrator to view any user's menu at any site. They said they are releasing jEM in 3 months as a "try before you buy" download. I am on the mailing list to get the release announcement.

LewisC 7/4/200

Phillipe,

I think the speed was the thing I meant to write about and forgot. I can't find my note but they have one user, military I assume, that has over 10,000 users and over 1000 menu options at a couple of hundred sites. That is impressive.

You can get the same kind of tree menu in Forms but it chokes under even a mildly high number of menu options.

The usability of the system, i.e. administration, was nice too. Everything was close to hand and very responsive. Once it's generally available, I may try to do a review.

Thanks for the post,

LewisC

TOAD User Group Conference/Meeting/Training

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/005650.asp

Posted 9/7/2005

I recently had the opportunity to attend a one-day TOAD user group in Tampa, FL. I was afraid it would either just be a marketing ploy or that it would be so basic that I wouldn't find it useful. In reality, I got a lot out of it. This was a conference for people who want to know what's coming in TOAD and who want to get the most out of TOAD.

I still use 8.0. I'm not sure when my company will upgrade to 8.5. The conference covered 8.5 so if I talk about something your version does not do, check the version you're running. For details on 8.0 vs. 8.5, check out this comparison.

I started using TOAD somewhere around 1996, 1997, something like that. It was a freeware tool for browsing Oracle databases. I've used a lot of other tools in that time but I always go back to TOAD. Quest did not own it at that time. I can't remember his name (if you do write me and let me know) but the tool was developed by a developer who spent a lot of time in Oracle and wanted a tool to help himself. I remember once writing to him and describing a bug in great detail. By way of thanks, he not only immediately fixed the bug; he emailed a new copy of TOAD to me. Heh Talk about support.

The conference was organized in seminar fashion. In the morning, everyone was in the same room and after lunch we split up into developer and DBA groups. Brian spoke during the morning and developer sessions; Dwayne spoke during the DBA session. I'm not going to cover everything they covered. I'll just hit the highlights.

Not all of these features are available in TOAD standard.

And a caveat, TOAD is very right click oriented. If you're looking for an action, try right clicking first. If you don't see it there go to the menus.

Many more people showed up than was expected. We had to open another room to have enough chairs for everyone. I imagine a couple of hundred people RSVPed but Quest expected only 50 or so to show up. I think there must have been 100 to 125 people there.

We started off with the multitude of configuration items in TOAD. Just about everything is configurable and more becomes configurable with each release. The options screen has a nice feature where you can search for an option. Say, for example, you would like to turn off the annoying "ribbit" sound that plays whenever you start up TOAD. You can enter WAV in the search field and any options that have WAV in their text are displayed. Double click on the option and the tab you need is opened. A nice little feature I rarely use but is nice to have.

Brian also covered how to switch the view in the schema browser from a tree view to a tab view. I prefer the tab view myself. The tree view will probably be preferred by the people who are most used to Microsoft products.

Brian covered a really nice feature, I think it must be 8.5, that will show you the SQL TOAD is executing behind the scenes. This could be a great time saver. The output window exists in my version of TOAD but only shows error messages and spooled output.

We also covered the SQL Modeler. I have used the modeler in the past. It's a nice query builder. It's a little but richer than I thought it was. I did not know that you could define calculated fields. This functionality reminded me a lot of BrioQuery.

The SQL Editor even had a few surprises for me. I spend the majority of my time in TOAD in three places: SQL Editor, Schema Browser and Session Browser. I thought I knew everything there was to know about the SQL Editor.

In the editor you can hit Ctrl+SpaceBar and a list of templates come up. Can't remember the syntax for a RECORD definition or an NDS query? Ctrl+SpaceBar. This is a really nice feature I totally did not know existed.

Also in the editor, you can use DESC F9 to see a table definition. Double click a column and it sends the column name to your editor. Nice.

The editor also features Auto Replace text. For example, if you're typing a select statement and type "SELECT * FORM DUAL", TOAD will automatically fix it as you type so that you get "SELECT * FROM DUAL".

After the SQL Editor we covered SQL Recall. I use this quite often but for those who don't; you can configure TOAD to remember all SQL Commands that are run (it defaults to the last 100 commands). You can also name SQLs and store them permanently.

TOAD also features the ability to generate code for different languages. It's called the Make Code facility. In the SQL Editor options (under View->Options), there is an option for Make Code Format and Make Code Variable Name. Choose the flavor from the drop down and then navigate to the SQL

Editor. Enter a SQL statement and press the little vb icon on the tool bar. It shows VB even if you chose a different language. Example:

I chose C++ and enter this: SELECT * FROM DUAL

I hit the VB icon and got this:

```
char SQL[18];
strcpy(SQL, "select * from dual");
```

That's handy. TOAD will do this for VB, Java, C++, Perl and Delphi.

On the flip side of this, if you have a statement already formatted for Java or C++ or whatever, use the Strip Code (the other VB icon) button and it will turn it back into regular SQL.

And finally, in the editor, in 8.5 and up, if run multiple commands using the "Run as Script" button, TOAD will open a grid for each query in the script. That's worth upgrading for all by itself.

This is getting much longer than I intended. Suffice it to say the conference, user group, meeting, whatever it was, was very much worth the time. And being free didn't hurt either. If you are a user of TOAD, I would recommend you attend this if given the opportunity. If you are not a user of TOAD, what are you doing reading this? Heh. If you aren't a user but are thinking about it, I would recommend this for you also.

If anyone is interested in me running through the rest of the day, let me know and I'll finish up. Some other things we covered were Script Manager, Schema Browser, Data Export and Import, Object and Schema Compares, Data Subset Wizard, Master/Detail Browser, HTML Schema Doc, Create like, TOAD and Report generation (nice, didn't know about this one and a full fledged report designer will be in 8.6), Project Manager, ERD Designer, TOAD FTP, SQL Loader wizard, Data Factory (another really nice tool), Benchmark Factory, Security Features in TOAD, Database Monitor, Top Session Monitor, Session Browser, DB Wizard, Knowledge Expert and SQL Tuning Lab.

As you can see it was a busy day.

Bottom line, if you can attend, you should!

TOAD User Group Conference/Meeting/Training, Part 2

Original Post: http://blogs.ittoolbox.com/oracle/guide/archives/005721.asp

Posted 9/12/2005

I'm picking up from where I left off in the previous posting on the first part of the TOAD User Group get together.

Next up was the import and export wizards. Nothing fancy but a nice to have feature that will help you get data in and out of the database.

Then we covered the Data Subset Wizard. Very nice. If you want to move a subset of your data from production to test or dev, the data subset wizard will grab a random subset of parent and child tables and move that data for you. The wizard uses foreign keys to maintain referential integrity. And notice that it is a RANDOM subset. If you need specific records or types of records for regression testing, this won't do it for you. It will fit the bill in many cases though. This is one feature that I didn't know even existed and it does exist in 8.0 Standard.

After that we spoke about the new flavors of TOAD. TOAD actually stands for Tool for Oracle Application Developers. I think going forward Quest will try to minimize the meaning of the acronym. Tool for Oracle Application Developers for DB2 just doesn't sound right. But Quest is making TOAD available for DB2, MySQL and SQLServer.

The speaker doing this part of the presentation got a lot of questions regarding this. From memory, the new flavors are written in either Java or C, I can't remember which. Oracle TOAD (OTOAD) is written in Delphi. The new flavors will maintain the same look and feel with each other but not necessarily with OTOAD. OTOAD will not be rewritten, at least not anytime soon, because the time it would take would be phenomenal. It will be a while before the new flavors support all of the features currently supported by OTOAD.

Several people asked about a version of TOAD for various other databases but at this time there are none planned.

Another topic that came up here, Quest bought TOra, a Linux Oracle browser like TOAD. The speaker did not know if there were any plans to update TOra or to make TOAD available for Linux.

There is also, finally, an update to the free version of TOAD. Unlike in the past though, this free version is crippleware. Disappointing in the extreme.

Next we covered TOAD's compare tools. Have you ever needed to compare one schema against another? Or a table in one schema against a table in another schema? Or how about comparing data in a table? TOAD will let you do all of those things. You can compare objects and data in the same database or in different databases via links. This is a nice feature that has been around for a while but has been updated some in 8.5. Check it out. Data Compare is found under the Tools menu and Schema Compare is found under the DBA menu.

We ran through the Master/Detail browser. This is pretty straight forward and from what I could tell has not been updated in a while. It's a very linear parent->child browser. Plans for the future are to allow multiple children per parent. That would be a nice addition.

I use an open source tool called OraSchemaDoc. It's a python utility to generate JavaDoc style documentation from an Oracle database. If you use comments on table and columns it's a really nice documentation tool. Anyway, TOAD has a feature very much like it called HTML Schema Doc Generator. It's very robust and configurable. I keep telling myself to switch but OraSchemaDoc has worked well for me up to this point.

We covered some of the delivered reports that come with TOAD. Personally, I don't like any of them. I have my own scripts that I use for the kinds of info they display. I did learn about the Print Preview designer. In the schema browser, open up a data grid and right click. Select Print Grid and then Preview. On the tool bar, select Design Report. In this designer, you can basically design a report on the fly. If you used a view as your dataset, you could do some fairly complicated things. For one off reports this could be a great tool. The biggest downside, no save. You cannot save your design. Supposedly a full feature report writer will be in 8.6.

The last thing we covered before lunch was the project manager. Basically this is a way to group project files together. You can even group ftp connections to the project just like files. A benefit of this is that you can save the password with the ftp connection. You can give away the project manager config file and people can ftp to the machine but since they don't know the password, they can't do anything else. I haven't tested this and would use it with caution.

The TOAD FTP client will support secure shell in 8.6. It's a nice little FTP client. It reminds me of WarFTP. I currently use FileZilla and don't think I'll switch any time soon. If you don't have a preferred client though, the TOAD FTP blows away the crappy thing that comes with Windows.

Quest provided us with lunch. Buffet style salad and sandwich. Good desserts. Of course, I ate too much and wanted to sleep through the second half of the day.

Speaking of which, for the second half of the day I decided to take the DBA track. The developer track covered debugging and such and I have used TOAD quite a bit for that. I may have been able to pick up a few tips but I thought I would get more out of the DBA track.

The first thing we covered after lunch was Data Factory. I love this tool. It is a test data generator. Quest supplies a set of tables with semi random data, or you can supply your own tables of data. You map a column in your application to a column in your data tables. The tool will randomly join up that data and insert it into your application tables. It's fairly robust and you can tell Data Factory what your keys are and it will automatically maintain referential integrity. Nice!

Next up was Benchmark Factory (BF). BF simulates multiple connections to your application for user load testing. This load test tests the database not the middle tier or client tiers. Its goal is to give an accurate idea of what your DATABASE can handle.

We covered making TOAD read-only by playing with the .LIC. Just copy READONLY.LIC to TOAD.LIC and TOAD is read-only. To return it to non-read-only, just copy FULLTOAD.LIC to TOAD.LIC.

You can turn features on and off for various users of TOAD. To do this you need to install database objects. I don't use this particular feature. For more info, go to Help and search on Security Features.

Towards the end of the day we quickly ran through the monitors. With the Database Monitor you get a health check of you database. It's like a mini-Spotlight (a real monitoring tool from Quest). We also looked at Top Session and the Session Browser.

I spend a lot of time in Session Browser. In here you can view currently executing SQL, IO and Long Ops, and even trace or kill sessions if you have the right permissions.

We very quickly covered the Database wizard that lets you create a database. Nice layout of options and such but I haven't personally used it. Great for beginners though.

We also briefly looked at knowledge expert (used to be revealnet) and SQL Tuning Lab. SQL Tuning Lab is a graphically oriented tool that offers best practices to tune your queries. You have to buy an add-on to get this feature.

And finally we covered the script manager. It's located under the File menu option. Script Manager allows you to group all of your frequently (or not so frequently) run scripts together in one place. You can run them via ScriptRunner or load it into a SQL Editor. TOAD comes with a lot of scripts already written.

Some of the scripts are nice as a learning tool. Read the scripts and see how they're put together.

And that is it for the TOAD Day in Tampa. For asking or answering certain questions, the speakers handed out the TOAD pocket reference. That's a nice little reference tool that covers TOAD through 8.5.

It is a far, far better thing that I do, than I have ever done; it is a far, far better rest that I go to than I have ever known." - Charles Dickens, A Tale of Two Cities

New Orleans, LA – Post Katrina